



**KÖŞE NOKTALARI ADLANDIRILMIŞ
AĞAÇLARIN SAYILARI**

Yüksek Lisans Tezi

Emre YİVLİ

Eskişehir 2023

KÖŞE NOKTALARI ADLANDIRILMIŞ AĞAÇLARIN SAYILARI

Emre YİVLİ

YÜKSEK LİSANS TEZİ

Matematik Anabilim Dalı Analiz ve Fonksiyonlar Teorisi Bilim Dalı

Danışman: Prof. Dr. Emrah AKYAR

Eskişehir

Eskişehir Teknik Üniversitesi

Lisansüstü Eğitim Enstitüsü

Nisan 2023

JÜRİ VE ENSTİTÜ ONAYI

Emre Yivli'nin "KÖŞE NOKTALARI ADLANDIRILMIŞ AĞAÇLARIN SAYILARI" başlıklı tezi 28/04/2023 tarihinde aşağıdaki jüri tarafından değerlendirilerek "Eskişehir Teknik Üniversitesi Lisansüstü Eğitim-Öğretim ve Sınav Yönetmeliği"nin ilgili maddeleri uyarınca, Matematik Anabilim Dalında Yüksek Lisans tezi olarak kabul edilmiştir.

<u>Jüri Üyeleri</u>	<u>Unvanı Adı Soyadı</u>	<u>İmza</u>
Üye	:	
Üye	:	
Üye	:	
Üye	:	
Üye	:	

Prof. Dr. Murat TANIŞLI
Lisansüstü Eğitim Enstitüsü Müdürü

28/04/2023

DANIřMAN ONAYI

Daniřmanlıęını yrttęm yksek lisans ęrencisi Emre Yivli ‘‘KēE NOKTALARI ADLANDIRILMIř AęAęLARIN SAYILARI’’ bařlıklı tez alıřmasını tamamlamıřtır. Hazırlamıř olduęu tez tarafımca incelenmiř ve ęrencinin tez savunması sınavına alınması bilimsel ve etik aıdan uygun grlmřtır.

Tez Daniřmanı
Prof. Dr. Emrah AKYAR

ÖZET

KÖŞE NOKTALARI ADLANDIRILMIŞ AĞAÇLARIN SAYILARI

Emre YİVLİ

Matematik Anabilim Dalı

Analiz ve Fonksiyonlar Teorisi Bilim Dalı

Eskişehir Teknik Üniversitesi, Lisansüstü Eğitim Enstitüsü, Nisan, 2023

Danışman: Prof. Dr. Emrah AKYAR

n köşe noktalı bir ağacın köşe noktaları $l_1, l_2, \dots, l_n$ gibi etiketlerle birbirinden ayrılabiliriyorsa böyle ağaçlara köşe noktaları adlandırılmış ağaçlar denir. İki ağaç izomorfik olsa bile köşe noktaları farklı adlandırılmışsa bu ağaçlar farklı ağaçlar olarak kabul edilir. Cayley teoremine göre n köşe noktalı köşe noktaları adlandırılmış n^{n-2} farklı ağaç vardır. Prüfer, n köşe noktalı ağaçlarla $n - 2$ uzunluğundaki 0 ile $n - 1$ arasındaki sayılardan oluşan dizilerin bire bir karşılık geldiğini göstererek bu formülün bir kanıtını vermiştir. Bu sayı dizilerine Prüfer dizisi denir. Prüfer dizileri ağaçların alışılagelmiş gösterimlerine bir alternatif sunar.

Bu çalışmada, bir bilgisayar cebir sistemi olan Maple için Prüfer dizilerinden faydalanılarak ağaçlarla ilgili prosedürler içeren bir kütüphane sunulmaktadır. Ayrıca, köşe noktaları adlandırılmış çift-yıldız, örümcek, kırkayak, vb. birçok özel ağacın sayıları Prüfer dizilerinden yararlanılarak kombinatorial yöntemlerle hesaplanmıştır.

Anahtar Sözcükler: Köşe noktaları adlandırılmış ağaçlar, Prüfer kodu, Bilgisayar cebir sistemi

ABSTRACT

NUMBER OF LABELLED TREES

Emre Yivli

Mathematics Department

Analysis and Functions Theory

Eskisehir Technical University, Institute of Graduate Programs, April, 2023

Supervisor : Prof. Dr. Emrah AKYAR

A tree with n vertices called a labelled tree if its vertices are distinguished from one another by names such as, $l_1, l_2, \dots, l_n$. Even if two trees are isomorphic, trees with having different vertex labels are considered as distinct graphs. According to Cayley's tree formula, there are n^{n-2} labelled trees on n vertices. Prüfer used a simple way to prove this formula and demonstrated that there exists a bijection between the set of labelled trees on n vertices and sequences of $n - 2$ numbers, each in the range $0, 1, 2, \dots, n - 1$. Such a number sequence is called a Prüfer code and it provides an alternative to the usual representation of trees.

In this study, a computer algebra system (Maple) library containing various algorithms for trees is presented with the help of Prüfer code. Moreover, the number of labelled trees for various families of trees such as double star, spider, centipede, firecracker, etc. is calculated using combinatorial methods with the help of Prüfer code.

Keywords: Labelled Tree, Prüfer Code, Computer Algebra System

TEŞEKKÜR

Tez çalışmam sırasında bilgi ve tecrübeleri ile bana yol gösterici ve destek olan değerli danışmanım sayın Prof. Dr. Emrah AKYAR'a teşekkür ve saygılarımı sunarım.

Çalışmalarım sırasında desteklerini esirgemeyen aileme teşekkür ederim.

Emre Yivli



ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ

Bu tezin bana ait, özgün bir çalışma olduğunu; çalışmamın hazırlık, veri toplama, analiz ve bilgilerin sunumu olmak üzere tüm aşamalarda bilimsel etik ilke ve kurallara uygun davrandığımı; bu çalışma kapsamında elde edilemeyen tüm veri ve bilgiler için kaynak gösterdiğimi ve bu kaynaklara kaynakçada yer verdiğimi; bu çalışmanın Eskişehir Teknik Üniversitesi tarafından kullanılan “bilimsel intihal tespit programı”yla tarandığını ve hiçbir şekilde “intihal içermediğini” beyan ederim. Herhangi bir zamanda, çalışmamla ilgili yaptığım bu beyana aykırı bir durumun saptanması durumunda, ortaya çıkacak tüm ahlaki ve hukuki sonuçlara razı olduğumu bildiririm.

Emre Yivli

İÇİNDEKİLER

	<u>Sayfa</u>
BAŞLIK SAYFASI	I
JÜRİ VE ENSTİTÜ ONAYI	II
DANIŞMAN ONAYI	III
ÖZET	IV
ABSTRACT	V
TEŞEKKÜR	VI
ETİK İLKE VE KURALLARA UYGUNLUK BEYANNAMESİ	VII
İÇİNDEKİLER	VIII
ŞEKİLLER DİZİNİ	XI
SİMGELER ve KISALTMALAR DİZİNİ	XII
1. GİRİŞ	1
1.1. Temel Kavramlar	3
2. AĞAÇLARI SAYMAK	8
2.1. Ağaçları Saymak	8
2.2. Bazı Özel Ağaçların Sayıları	15
2.3. Rastgele Ağaçlar Üretmek	24
3. AĞAÇLAR İÇİN MAPLE PROSEDÜRLERİ	25
3.1. Komut Adı: Prufer2Tree	25
3.2. Komut Adı: Tree2Prufer	26
3.3. Komut Adı: RandomTree	26
3.4. Komut Adı: GetTree	27
3.5. Komut Adı: DoubleStar	27
3.6. Komut Adı: Centipede	28
3.7. Komut Adı: FireCracker	28
3.8. Komut Adı: BananaTree	29
3.9. Komut Adı: Caterpillar	29
3.10. Komut Adı: SemiPath	30

3.11.Komut Adı: Comet	30
3.12.Komut Adı: Spider	31
3.13.Komut Adı: CayleyTree	31
3.14.Komut Adı: Tripod	32
EK MAPLE KODLARI	33
KAYNAKÇA	37
ÖZGEÇMİŞ	



ŞEKİLLER DİZİNİ

	<u>Sayfa</u>
Şekil 1.1. Bir basit G çizgesi	3
Şekil 1.2. Maple komutlarıyla oluşturulmuş bir çizge	4
Şekil 1.3. Maple komutlarıyla oluşturulmuş Petersen çizge	4
Şekil 1.4. Maple komutlarıyla oluşturulmuş rastgele bağlantılı bir çizge	4
Şekil 1.5. Köşe noktaları adlandırılmış bir çizge	5
Şekil 1.6. Bir çizge üzerinde tur ve yol farkı	5
Şekil 1.7. Sırasıyla bağlantılı ve bağlantısız olan iki çizge	6
Şekil 1.8. İzomorfik G_1 ve G_2 çizgeleri ve izomorfik olmayan G_3 çizgesi	6
Şekil 2.9. Üç farklı ağaç	8
Şekil 2.10. 5 köşe noktalı köşe noktaları adlandırılmamış tüm farklı ağaçlar	9
Şekil 2.11. Köşe noktaları adlandırılmış bir ağaç örneği	9
Şekil 2.12. Kökü “0” ile adlandırılmış köklü bir ağaç	10
Şekil 2.13. Şekil 2.12 ile verilen ağacın Prüfer dizisinin bulunması	11
Şekil 2.14. Köşe noktaları adlandırılmış bir ağaç	12
Şekil 2.15. Şekil 2.14 ile verilen ağaca Algoritma 2.5’nin uygulanması	12
Şekil 2.16. İterasyon 1 ve 2	14
Şekil 2.17. İterasyon 3 ve 4	14
Şekil 2.18. İterasyon 5 ve oluşan ağacın farklı bir gösterimi	14
Şekil 2.19. İzomorfik olmayan ve aynı derece dizilerine sahip iki ağaç	14
Şekil 2.20. Kökü “0” olan ve köşe noktaları adlandırılmış P_4 patika çizge	15
Şekil 2.21. Kökü “0” olan ve köşe noktaları adlandırılmış yıldız çizge	15
Şekil 2.22. $F_{4,4}$ havai fişek çizge	16
Şekil 2.23. 5-kırkayak ağaç	17
Şekil 2.24. $S_{7,8}$ çift yıldız ağaç	17
Şekil 2.25. 14 köşe noktalı ve 8 yapraklı örümcek ağaç	18
Şekil 2.26. $(5,4)$ —muz ağacı	19
Şekil 2.27. 11’inci dereceden 4—kuyruklu yıldız çizge	20
Şekil 2.28. 9 köşe noktalı bir tırtıl çizge	20
Şekil 2.29. Üç farklı 3-Cayley ağacı	21
Şekil 2.30. Kökü “0” olan bir 3-Cayley ağacı	21
Şekil 2.31. $(6,2)$ -yıldız çizge	22
Şekil 2.32. O_3 zeytin ağacı	23
Şekil 2.33. Sırasıyla 5 ve 6 köşe noktalı iki yarı patika çizge	23
Şekil 2.34. Üç farklı tripod ağaç	24

Şekil 2.35. Prüfer dizisi (0,0,1,2,3,4) olan ağaç	25
Şekil 3.36. Prufer2Tree komutu ile oluşturulmuş bir ağaç	26
Şekil 3.37. T ağacının Tree2Prufer komutu ile oluşturulmuş kodu	26
Şekil 3.38. RandomTree komutu ile oluşturulmuş rastgele ağaç	27
Şekil 3.39. GetTree komutunun uygulanışı	27
Şekil 3.40. DoubleStar komutunun uygulanışı	28
Şekil 3.41. Centipede komutunun uygulanışı	28
Şekil 3.42. FireCracker komutunun uygulanışı	29
Şekil 3.43. BananaTree komutunun uygulanışı	29
Şekil 3.44. Caterpillar komutunun uygulanışı	30
Şekil 3.45. SemiPath komutunun uygulanışı	30
Şekil 3.46. Comet komutunun uygulanışı	31
Şekil 3.47. Spider komutunun uygulanışı	31
Şekil 3.48. CayleyTree komutunun uygulanışı	32
Şekil 3.49. Tripod komutunun uygulanışı	32

SİMGELER ve KISALTMALAR DİZİNİ

V	: Çizgenin köşe noktaları kümesi
E	: Çizgenin kenarlar kümesi
$G = (V, E)$	: Köşe noktaları kümesi V ve kenarlar kümesi E olan çizge
$\deg(v)$	: v köşe noktasının derecesi
$v_0 \rightarrow v_1 \rightarrow \cdots \rightarrow v_{k-1} \rightarrow v_k$	: v_0 ve v_k köşe noktalarını birleştiren yolculuk
T	: Ağaç
P_n	: n köşe noktalı patika çizge
S_n	: n köşe noktalı yıldız çizge
$F_{n,k}$	: Havai fişek çizge
c_n	: Kırkayak çizge
$S_{r,s}$	: Çift yıldız çizge
O_k	: Zeytin ağacı
$[x]$	: x gerçel sayısının tam kısmı

1. GİRİŞ

Çizge, farklı objeler ve bu objeler arasındaki ilişkileri veren matematiksel bir yapıdır. Objeler, çizgenin köşe noktaları, bu objeler arasındaki ilişkiler ise çizgenin kenarları olarak adlandırılır. Çizge kuramı, matematiğin nispeten yeni bir alanıdır ve çizge kuramının 1735 yılında ünlü matematikçi Leonhard Euler'in çalışmaları ile başladığı kabul edilir. Çizge kuramı o zamandan beri, bilimin hemen hemen her alanında kullanılan güçlü bir araca dönüşmüştür ve şu anda matematiğin en aktif araştırma alanlarından birisidir. Ağaçlar, bağlantılı ve döngü içermeyen özel tipte çizgelerdir. Ağaçlar, Alman fizikçi Gustav Kirchhoff'un 1847'de elektrik ağlarındaki akımlarla ilgili çalışmasında dolaylı olarak ortaya çıkarken, Arthur Cayley 1857'de belirli kimyasal bileşik türlerini saymak için ağaçları kullanmıştır ([1, 2]). Ağaçlar, başta bilgisayar biliminde verileri organize etmek ve kullanmak için bir araç olarak kullanılmakla birlikte birçok alanda yaygın olarak kullanılmaktadır. Belli sayıda köşe noktasına sahip, izomorfik olmayan ağaçların sayısı için kapalı bir formül olmamasına rağmen, n köşe noktalı, köşe noktaları adlandırılmış ağaçların sayısı için bir formül mevcuttur. Köşe noktalarının sayısı n olan, köşe noktaları adlandırılmış n^{n-2} farklı ağaç vardır. Bu formül 1889 yılında ünlü İngiliz matematikçi Arthur Cayley tarafından keşfedilirken bu çalışmada ele alacağımız kanıt, Abelian grupları üzerine yaptığı çalışmalarla tanınan Alman matematikçi Heinz Prüfer tarafından 1918'de verilmiştir ([3]). Prüfer kanıtında n köşe noktalı, köşe noktaları adlandırılmış ağaçlar ile bileşenleri n farklı sayıdan oluşan $n - 2$ bileşenli diziler arasında birebir bir eşleme vermiştir. Böylece bu tür dizilerin sayısını hesaplayarak Cayley'in köşe noktaları adlandırılmış ağaçların sayısını veren formülüne ulaşmıştır. Bu nedenle sözü edilen dizilere Prüfer dizisi adı verilir. Ağaçları saymak, çizge kuramının üzerinde çok çalışılan önemli konularından birisi olmakla beraber bugüne kadar birçok ağaç ailesi için çeşitli formüller elde edilmiştir ([4]).

Maalesef çizge kuramına ait birçok problem köşe noktaların sayısı arttığında elle hesaplanamayacak zorluğa ulaşır. Bu durumda bir bilgisayar cebir sisteminden yardım almak kaçınılmaz olur. Bu çalışmada kullanım kolaylığı ve yaygın olarak kullanılması nedeniyle Maple tercih edilmiştir. Maple, Waterloo, Kanada'da kurulmuş bir yazılım şirketi olan Maplesoft tarafından geliştirilen bir bilgisayar cebir sistemidir. Maple, ilk olarak 1980 yılında Waterloo Üniversitesi'ndeki sembolik hesaplama grubu tarafından kurulmuştur [5]. Yazılım, grup teorisi paketi, çizge kuramı paketi, lineer cebir paketi gibi birçok paketi içermektedir. Bu çalışmada temel olarak kullanılan çizge kuramı paketi Simon Fraser Üniversitesi'nde yüksek lisans öğrencileri tarafından geliştirilmiştir. Paketin planı ilk olarak Waterloo'daki 2005 Maple konferansında tanıtılmış, iyileştirmeler ise 2006 Maple konferansında sunulmuştur. Çizge kuramı paketinin ilk

sürümü Maple 11 sürümünde 2007 yılında GraphTheory paketi olarak yayınlanmıştır ([6]). 2007 yılından bu yana da paketin içeriğinde birçok iyileştirme yapılmış ve RandomGraphs, SpecialGraphs gibi bazı alt paketler de eklenmiştir. GraphTheory paketi, genel olarak çizgeleri oluşturmak, çizmek, üzerlerinde değişiklik yapmak ve çeşitli özelliklere sahip olup olmadıklarını test etmek için kullanılan bir rutinler koleksiyonudur. Ancak henüz ağaçlar ile ilgili yeterince gelişme kaydedilememiştir.

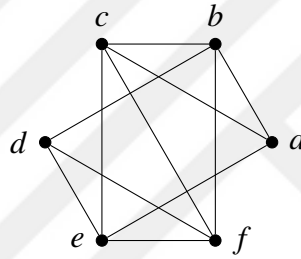
Üç bölümden oluşan bu çalışmanın ilk bölümünde çizgeler ile ilgili temel kavramlar verilmiştir. İkinci bölümde ise önce ağaçlar ile ilgili temel kavramlar ve ağaçların çeşitli özellikleri sıralanmıştır. Daha sonra çift-yıldız, kırkayak, örümcek gibi bazı özel ağaçlar tanıtılmıştır. Aynı bölüm içerisinde bu özel ağaçların sayıları için Prüfer dizileri yardımıyla köşe nokta sayısına bağlı formüller sunulmuştur. Son olarak üçüncü bölümde çeşitli ağaç aileleri için Prüfer dizilerinden yararlanılarak oluşturulan bir Maple kütüphanesi sunulmuştur.

1.1. Temel Kavramlar

Burada sonraki bölümler için gerekli olan çizgelerin bazı önemli özelliklerinden bahsedilecektir.

Tanım 1.1. Bir G çizgesi, elemanları köşe noktası olarak adlandırılan, boş kümeden farklı, sonlu bir V kümesiyle, elemanları kenar olarak adlandırılan ve V kümesinin iki elemanlı alt kümelerinden oluşan E kümesi yardımıyla tanımlanır. Bu durumda G çizgesi $G = (V, E)$ şeklinde yazılır.

Bu çalışmada yalnızca basit çizgeler incelenmiştir. Basit çizgeler; ağırlıksız, yönsüz, bir köşe noktasından kendisine bir kenar bulundurmeyen ve iki köşe noktası arasında en fazla bir kenar bulunduran çizgelerdir. Çizgeler çoğunlukla çizgenin köşe noktaları düzlemde birer nokta, kenarları ise bu noktalar arasına çizilen kenarlar olmak üzere, düzlemde çizilen diyagramlar yardımıyla verilir.



Şekil 1.1. Bir basit G çizgesi

Bir G çizgesinin u ve v köşe noktalarını birleştiren bir $\{u, v\}$ kenarı varsa bu köşe noktalarına komşu köşe noktaları denir.

Örneğin, Şekil 1.1 ile verilen G çizgesi için

$$V = \{a, b, c, d, e, f\}$$

ve

$$E = \{\{a, b\}, \{a, c\}, \{a, e\}, \{b, c\}, \{b, d\}, \{b, f\}, \{c, e\}, \{c, f\}, \{d, e\}, \{d, f\}, \{e, f\}\}$$

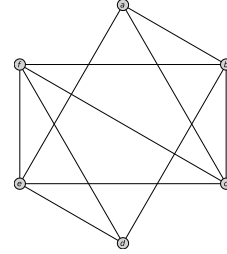
olur. Bu G çizgesi için $\{a, b\} \in E$ olduğu için a ve b komşudur.

Birçok bilgisayar cebir sisteminde çizge kuramı ile ilgilenenlerin kullanımına yönelik çeşitli kütüphaneler oluşturulmuştur. Maple programında ise GraphTheory paketi çizgeleri oluşturmak, çizmek, yönlendirmek ve belirli koşulları sağlayıp sağlamadığını test etmek için oluşturulmuş prosedürler içerir. [7]

Maple yazılımında basit çizge oluşturmak için kullanılan ana komut Graph

komutudur. DrawGraph komutu ile oluşturulan çizge çizdirilebilir. Örneğin, aşağıda verilen komutlar Şekil 1.1 ile verilen çizgeyi tanımlayıp çizdirmek için kullanılmıştır.

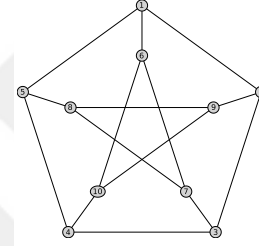
```
with(GraphTheory):
V := [a,b,c,d,e,f];
E := { {a,b},{a,c},{a,e},{b,c},
        {b,d},{b,f},{c,e},{c,f},
        {d,e},{d,f},{e,f}};
G := Graph(V,E);
DrawGraph(G);
```



Şekil 1.2. Şekil 1.1 ile verilen çizgenin maple komutlarıyla oluşturulmuş çıktısı

GraphTheory[SpecialGraphs] alt paketi ise Petersen çizge, tekerlek çizge, küp çizge gibi birçok özel çizgeyi içerir.

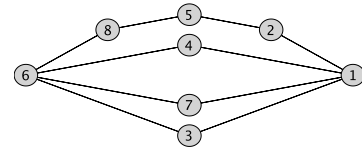
```
with(GraphTheory):
with(SpecialGraphs):
G := PetersenGraph();
DrawGraph(G);
```



Şekil 1.3. Maple komutlarıyla oluşturulmuş Petersen çizge

Son olarak GraphTheory[RandomGraphs] alt paketi ise rastgele çizgeler üretmek için kullanılır. Örneğin, köşe noktaları kümesi $\{1,2,\dots,8\}$ ve 10 kenara sahip bağlantılı rastgele bir çizgeyi aşağıdaki Maple komutları yardımıyla üretip çizdirilebilir.

```
with(GraphTheory):
with(RandomGraphs):
G:=RandomGraph(8,10,connected);
DrawGraph(G,style=spring);
```



Şekil 1.4. Maple komutlarıyla oluşturulmuş rastgele bağlantılı bir çizge

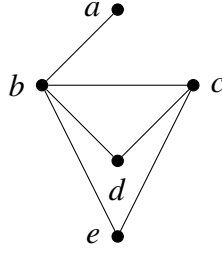
Tanım 1.2. Bir G çizgesinde v köşe noktasından çıkan kenar sayısına v köşe noktasının derecesi denir ve $\deg(v)$ şeklinde gösterilir.

Örneğin, Şekil 1.5 ile verilen çizgede a noktasının derecesi 1, d ve e noktalarının derecesi 2, c noktalarının derecesi 3 ve b noktasının derecesi ise 4'tür.

Teorem 1.3. (El sıkışma teoremi) Bir çizgede köşe noktalarının dereceleri toplamı kenar sayısının iki katını verir.

Sonuç 1.4. Bir çizgede tüm köşe noktalarının dereceleri toplamı çift sayıdır.

Sonuç 1.5. Bir çizgede derecesi tek olan köşe noktalarının sayısı çifttir.



Şekil 1.5. Köşe noktaları adlandırılmış bir çizge

Tanım 1.6. Bir $G = (V, E)$ çizgesinde $v_0, v_1, \dots, v_k \in V$ olmak üzere

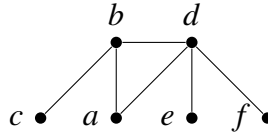
$$v_0 v_1, v_1 v_2, \dots, v_{k-1} v_k$$

şeklinde ardışık kenarlar bitişik ya da özdeş olacak biçimdeki kenarların sonlu dizisine G içinde v_0 ve v_k köşe noktalarını birleştiren bir yolculuk denir. v_0 ve v_k köşe noktalarını birleştiren yolculuk $v_0 \rightarrow v_1 \rightarrow \dots \rightarrow v_{k-1} \rightarrow v_k$ şeklinde de gösterilir.

Bir yolculuktaki kenar sayısına o yolculuğun uzunluğu denir.

Bu tanımlanan yolculuk kavramı oldukça genel bir kavramdır. Bu kavram üzerine koyulan bazı kısıtlarla ileride sıkça kullanılacak olan aşağıdaki temel kavramlar elde edilir.

Tanım 1.7. G çizgesindeki bir yolculuğun tüm kenarları farklı ise bu yolculuğa bir tur denir. Eğer turun tüm köşe noktaları farklı ise bu tura da bir yol adı verilir.



Şekil 1.6. $c \rightarrow b \rightarrow a \rightarrow b$ yolculuğu, c köşe noktasından b köşe noktasına bir tur değildir. $c \rightarrow b \rightarrow d \rightarrow a \rightarrow b$ yolculuğu ise, a ve b köşe noktalarını birleştiren bir tur olmasına karşın bir yol değildir

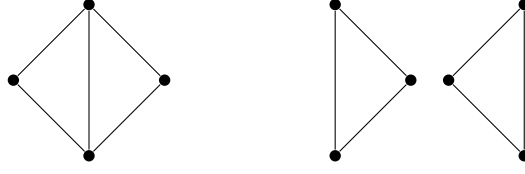
Tanım 1.8. Bir yolun başlangıç ve bitiş noktaları aynı ise bu yola bir döngü adı verilir.

Örneğin, Şekil 1.6 ile verilen çizgede $a \rightarrow d \rightarrow b \rightarrow a$ yolculuğu döngüdür.

Tanım 1.9. Bir G çizgesinde herhangi iki köşe noktasını birleştiren bir yolculuk var ise G çizgesine bağlantılıdır denir.

Örneğin Şekil 1.7 ile bağlantılı ve bağlantısız iki farklı çizge örneği verilmiştir.

Bir G çizgesi çok farklı şekillerde çizilebilir. Çizge çizilirken köşe noktalarının düzlemde nasıl konumlandırılacağı ya da çizgenin kenarlarının düz çizgilerle mi yoksa eğrilerle mi çizileceği, hatta bu kenarların birbiriyle kesişip kesişmeyeceği önemli değildir. Dolayısıyla “aynı” çizge farklı şekillerde çizilebilir.



Şekil 1.7. Sırasıyla bağlantılı ve bağlantısız olan iki çizge

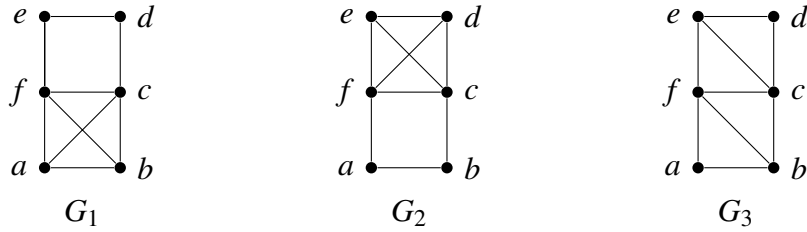
Kimi zaman verilen iki çizge aynı olmasa da çizgelerden birinin köşe noktalarının adları silinip, köşe noktalar uygun şekilde yeniden adlandırılarak çizgeler aynı hale getirilebilir. Bu tür çizgelere izomorfik çizgeler denir. Daha kesin olarak aşağıdaki tanım verilebilir

Tanım 1.10. $G_1 = (V_1, E_1)$ ve $G_2 = (V_2, E_2)$ çizgeleri verilsin. Eğer

$$\begin{array}{ccc} a, b \in V_1 \text{ köşe noktaları, } G_1 \text{ çizgesinde} & \Leftrightarrow & \phi(a), \phi(b) \in V_2 \text{ köşe noktaları, } G_2 \\ \text{komşu köşe noktalar} & & \text{çizgesinde komşu köşe noktalar} \end{array}$$

olacak şekilde bir $\phi : V_1 \rightarrow V_2$ birebir dönüşümü varsa, G_1 ve G_2 çizgelerine izomorfik çizgeler denir ve bu durum $G_1 \cong G_2$ şeklinde gösterilir. Bu şekildeki ϕ dönüşümüne ise G_1 ile G_2 arasında bir izomorfizm dönüşümü denir.

Örneğin Şekil 1.8 ile verilen üç farklı çizgeden ilk ikisi birbiri ile izomorfik iken diğer çizge bu iki çizgeye izomorfik değildir.



Şekil 1.8. Birbirine izomorfik olan G_1 ve G_2 çizgeleri ile bunlara izomorfik olmayan G_3 çizgesi

Şekil 1.8 ile verilen üç çizgenin de 6 noktası, 9 kenarı var ve her üçünün de derece dizileri $(4, 4, 3, 3, 2, 2)$ olmak üzere aynıdır. Bununla birlikte $G_1 \cong G_2$ olur. Çünkü

$$\begin{array}{l} a \leftrightarrow d \\ b \leftrightarrow e \\ c \leftrightarrow c \\ d \leftrightarrow b \\ e \leftrightarrow a \\ f \leftrightarrow f \end{array}$$

eşlemesi G_1 'in köşe noktaları ile G_2 'nin köşe noktaları arasında tanıtmda verilen koşulları sağlayan bir eşlemedir. Öte yandan $G_1 \not\cong G_3$ olur çünkü tümleyenlerine bakılırsa birinde üçgenler varken diğesinde yoktur.

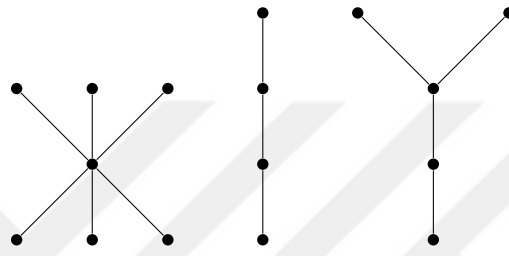


2. AĞAÇLARI SAYMAK

Özel bir çizge ailesi olan ağaçlar, çizgelerin önemli bir sınıfını oluşturur. Başta teorik bilgisayar bilimleri olmak üzere geniş uygulama alanları nedeniyle ağaçların önemi her geçen gün daha da artmaktadır. Bu bölümde tezin temel kavramı olan ağaçların çeşitli özellikleri verilmektedir. Ağaçlar ve ağaçların uygulamaları ile ilgili daha fazla bilgi edinmek için [8, 9, 10, 11] referanslarından yararlanılabilir.

Tanım 2.1. *Döngü içermeyen bağlantılı çizgelere ağaç denir.*

Ağaçlarla çalışırken bir ağacı belirtmek için genellikle G yerine T sembolü kullanılır.



Şekil 2.9. Üç farklı ağaç

Şekil 2.9 ile üç farklı ağaç örneği verilmektedir.

Aşağıda ağaçlarla ilgili iki önemli sonuç verilmektedir.

Teorem 2.2. *En az iki köşe noktası olan bir ağacın en az iki yaprağı vardır.*

Teorem 2.3. *n köşe noktalı bir ağacın kenar sayısı $n - 1$ olur.*

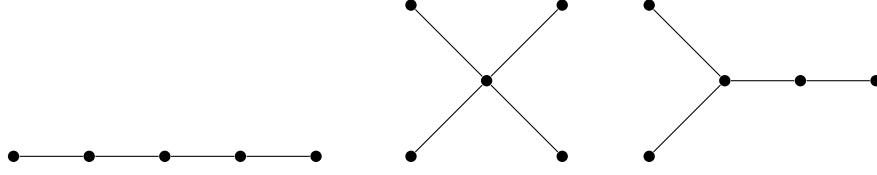
2.1. Ağaçları Saymak

Az sayıda köşe noktasına sahip olan ağaçların sayıları kolaylıkla bulunabilir. Örneğin 1 ya da 2 köşe noktalı köşe noktaları adlandırılmış sadece 1 ağaç vardır. Öte yandan bu sayı köşe noktaları adlandırılmamış ağaçlar için de geçerlidir. Diğer taraftan 3 köşe noktalı ağaçlar P_3 patika çizgeye (Bkz. Tanım 2.12) izomorfiktir. Bu şekilde köşe noktaları adlandırılmış toplam 3 farklı ağaç vardır ancak köşe noktaları adlandırılmamış yalnızca bir ağaç vardır. P_3 patika çizgenin köşe noktaları $\frac{3!}{2}$ farklı şekilde adlandırılabilir.

Benzer olarak, köşe noktaları adlandırılmış 4 köşe noktalı tüm farklı ağaçların sayısı da hesaplanabilir. 4 köşe noktalı tüm ağaçlar P_4 patika çizgeye ya da S_4 yıldız çizgeye izomorfik olur. Dolayısıyla köşe noktaları adlandırılmamış 2 farklı ağaç vardır. P_4 patika çizgenin köşe noktaları $\frac{4!}{2}$ farklı şekilde adlandırılabilirken S_4 yıldız çizgenin köşe noktaları 4 farklı şekilde adlandırılabilir. O halde 4 köşe noktalı köşe noktaları adlandırılmış $12 + 4 = 16$ farklı ağaç vardır.

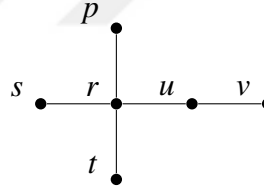
Görüldüğü üzere köşe noktalarının sayısı arttıkça köşe noktaları adlandırılmamış farklı ağaçların sayısı ile köşe noktaları adlandırılmış farklı ağaçların sayıları arasındaki fark

artmaktadır. Örneğin Şekil 2.10 ile köşe noktaları adlandırılmamış tüm farklı ağaçlar verilmiştir. Ancak 5 köşe noktalı köşe noktaları adlandırılmış farklı ağaçların sayısı 125'tir.



Şekil 2.10. 5 köşe noktalı köşe noktaları adlandırılmamış tüm farklı ağaçlar

Ağaçlarda derecesi 1 olan köşe noktalarına ağacın yaprakları denir. Örneğin Şekil 2.11 ile verilen ağacın yaprakları p, s, t, v köşe noktaları olur. Bir T ağacı verildiğinde çoğu zaman belirli bir köşe noktası “kök” olarak isimlendirilir. Bu tür ağaçlara köklü ağaç adı verilir. Bir T ağacı verilsin ve r köşe noktası bu ağacın kökü olarak seçilsin. v köşe noktası da T ağacının keyfi bir köşe noktası olarak belirlensin. Ağaç tanımı gereğince v köşe noktasından r köşe noktasına yalnızca bir yol vardır. v köşe noktasından r köşe noktasına bu yol üzerinden giderken karşılaşılan ilk köşe noktası v noktasının babası olarak adlandırılır. Örneğin Şekil 2.11 ile verilen ağaçta kök r olarak alınırsa v köşe noktasının babası u köşe noktası olur.



Şekil 2.11. Köşe noktaları adlandırılmış bir ağaç örneği

Çizge kuramına birçok katkıda bulunmuş olan Alman matematikçi Ernst Paul Heinz Prüfer, köşe noktaları $0, 1, \dots, n-1$ şeklinde adlandırılmış n köşe noktalı bir ağacın $n-2$ uzunluğunda ve $\{0, 1, \dots, n-1\}$ elemanlarından oluşan bir diziye karşılık getirilebileceğini, üstelik ağaçlarla bu tür diziler arasındaki eşlemenin birebir bir eşleme olduğunu göstermiştir [3]. Bu şekilde oluşturulabilecek tüm farklı dizilerin sayısı n^{n-2} olduğundan köşe noktaları adlandırılmış n köşe noktalı n^{n-2} tane ağaç olduğu söylenebilir. Bu ise meşhur Cayley Teoremidir.

Teorem 2.4 (Cayley Teoremi). [12] n köşe noktalı ve köşe noktaları adlandırılmış n^{n-2} ağaç vardır.

Bir T ağacının Prüfer dizisi aşağıdaki şekilde elde edilir:

1. Adım: Köşe noktalarının sayısı n olan T ağacının köşe noktaları $0, 1, 2, \dots, n-1$ şeklinde adlandırılınsın ve “0” ile adlandırılan köşe noktası kök olsun.

2. Adım: Ağacın kök hariç, derecesi 1 olan, en küçük sayı ile adlandırılan köşe noktası babası ile birlikte ele alınsın. Örneğin, Şekil 2.12 ile verilen ağaçta derecesi 1 olan köşe noktası “2” ile adlandırılan köşe noktasıdır ve bu köşe noktasının babası “1” ile adlandırılan köşe noktasıdır.

3. Adım: 2. Adımda bulunan, derecesi 1 olan en küçük sayı ile adlandırılan köşe noktası üst satırda ve babası alt satırda olmak üzere iki satırlık bir sütun şeklinde yan yana yazılır. Örneğin, Şekil 2.12 göz önünde bulundurulursa derecesi 1 olan ve en küçük sayı ile adlandırılan köşe noktalarından en küçük etiket ile adlandırılan köşe noktası “2” ve bu köşe noktasının babası ise “1” ile adlandırılan köşe noktasıdır. Bu durumda baba alt satırda ve kendisi üst satırda olmak üzere 2 satırlık bir sütun şeklinde yazılırsa

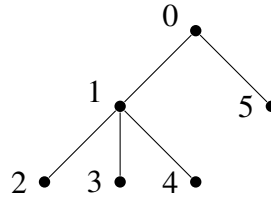
2 → derecesi 1 olan en küçük etiketli köşe noktası

1 → babası

şeklinde olur.

4. Adım: Derecesi 1 olan köşe noktaları arasında etiketi en küçük olan köşe noktası ve onun bağlı olduğu kenar T ağacından silinir. Bu şekilde elde edilen ağaç için 2. Adımdan başlanarak aynı adımlar köke ulaşınca kadar tekrar edilir.

Dolayısıyla Şekil 2.12 ile verilen ağaç için “2” ile adlandırılmış köşe noktası ve bağlı bulunduğu kenar ağaçtan silinerek 2. Adıma dönüp köke ulaşınca kadar devam edilir.

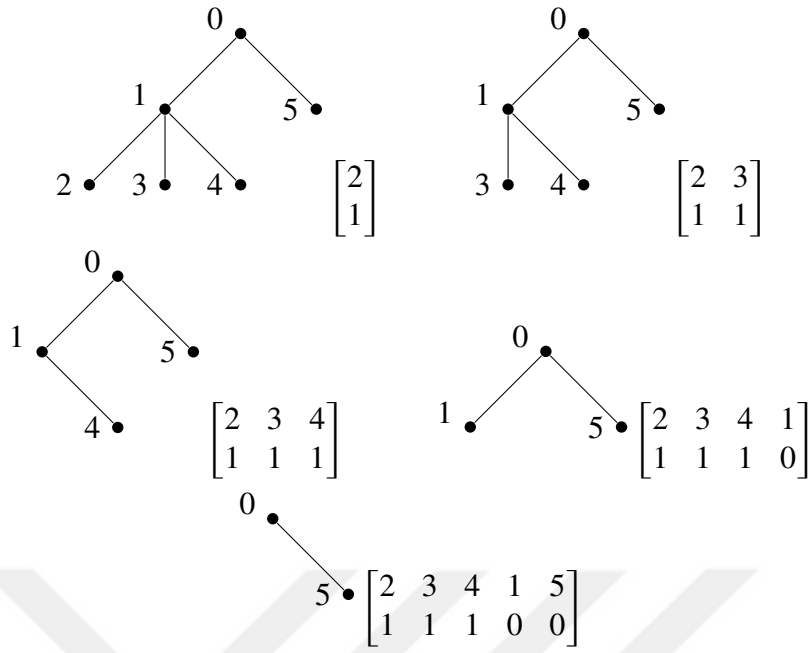


Şekil 2.12. Kökü “0” ile adlandırılmış köklü bir ağaç

Yukarıdaki işlem bir ağacın yapraklarından başlanılarak budanması gibi düşünülebilir. Bu işlem en küçük etiketli yapraktan budamaya başlayarak köke ulaşınca kadar devam edecektir. O halde, n köşe noktalı bir ağacın $n - 1$ kenarı olduğundan işlem sona erdiğinde $2 \times (n - 1)$ boyutlarında bir matris elde edilecektir.

Bu yöntem ile sona kalan köşe noktası kök olacağı için matrisin 2. satır $n - 1$. sütunundaki eleman her zaman “0” olacaktır. Bu yöntemle elde edilen matrisin 2. satırındaki ilk $n - 2$ sütunundaki elemanların dizisine Prüfer dizisi denir. Dolayısıyla, Şekil 2.12 ile verilen ağacın Prüfer dizisi Şekil 2.13 ile verilen hesaplamaya göre (1, 1, 1, 0) olur.

Böylece n köşe noktalı her köklü ağaca terimleri 0 ve $n - 1$ arasındaki sayılar olmak üzere $n - 2$ uzunluğunda bir dizinin karşılık getirilebileceği görülmüş oldu. Elbette aynı ağaca farklı iki dizinin karşılık gelmesi verilen yöntem gereğince mümkün değildir.



Şekil 2.13. Şekil 2.12 ile verilen ağacın Prüfer dizisinin bulunması

Yukarıda anlatılan işlemler bir algoritma şeklinde aşağıdaki gibi ifade edilebilir.

Algoritma 2.5.

Girdi $T = (V, E)$ ağacı.

Çıktı Prüfer dizisi $P = (v_1, v_2, \dots, v_{n-2})$.

Adım 1 1 ile $n - 2$ arasındaki her i tamsayısı için yap

başla

$u_i \leftarrow T$ ağacının en küçük etiketli yaprağı ($u_i \neq \text{kök}$)

$v_i \leftarrow u_i$ köşe noktasının komşusu

$T \leftarrow T - u_i$

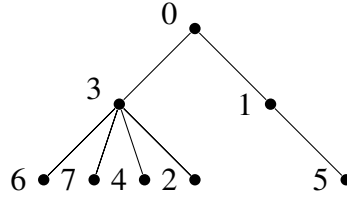
bitir

Adım 2 çıktı $(v_1, v_2, \dots, v_{n-2})$

Algoritma 2.5, doğrusal bir algoritma olmasa da literatürde bir ağacın Prüfer kodunu veren doğrusal algoritmalar da mevcuttur ([13, 14]).

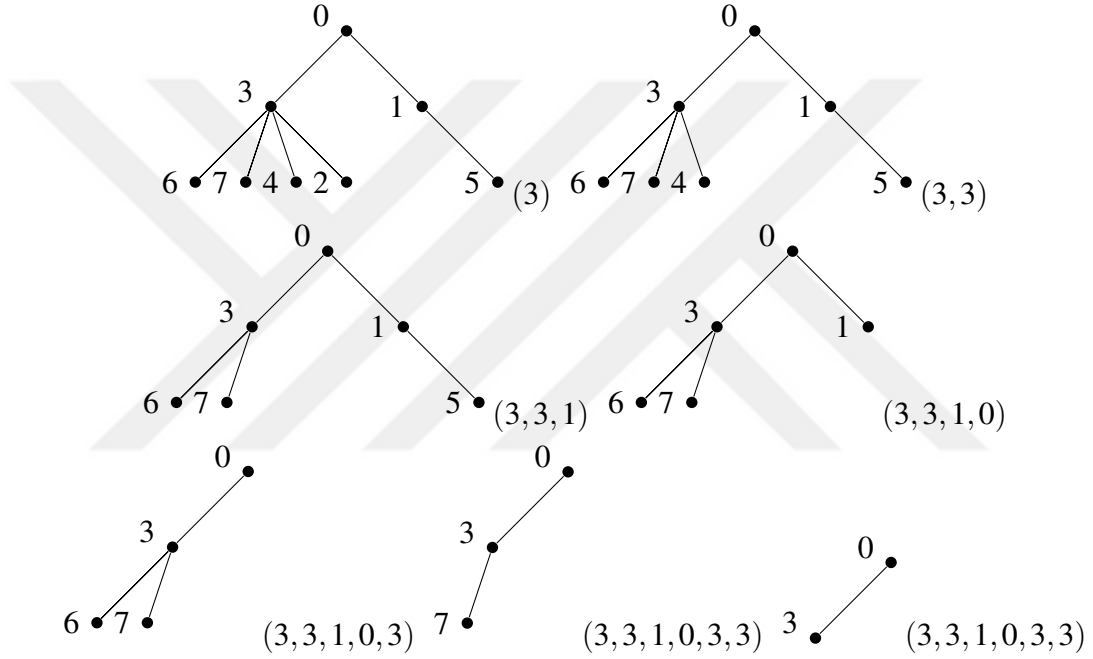
Örnek 2.6. Şekil 2.14 ile verilen ve kökü “0” olan köşe noktaları adlandırılmış ağacın Prüfer dizisi Algoritma 2.5 yardımıyla aşağıdaki gibi bulunabilir.

Şekil 2.14 ile verilen ağacın en küçük etiketli yaprağı “2” ile adlandırılmış köşe noktasıdır. Dolayısıyla “2” ile adlandırılmış köşe noktasının komşusu olan köşe noktasının etiketi



Şekil 2.14. Köşe noktaları adlandırılmış bir ağaç

“3” olduğundan Prüfer dizisinin ilk terimi “3” olur. Daha sonra “2” ile adlandırılmış köşe noktası ve $\{2,3\}$ kenarı ağaçtan silinir ve ilk adıma dönülür. Tüm adımlar Şekil 2.15 ile verilmiştir.



Şekil 2.15. Şekil 2.14 ile verilen ağaca Algoritma 2.5'nin uygulanması

Önerme 2.7. Prüfer dizisi matrisin tamamını verir.

Gerçekten, soldan başlayarak matrisin ilk satırındaki her girdi üst satırda daha önce geçmemiş ikinci satırda kendi altında ve sağında görülmeyen en küçük etiketli girdi olur. Bu da aslında bize ilk satırın nasıl elde edilebileceğini gösterir. Bundan böyle Prüfer dizisi verilen n köşe noktalı bir ağacı oluşturmak için aşağıda verilen algoritma kullanılacaktır.

Algoritma 2.8.

Girdi $1 \leq i \leq n - 2$ için $v_i \in \{0, 1, 2, \dots, n - 1\}$ olmak üzere Prüfer dizisi $P = (v_1, v_2, \dots, v_{n-2})$.

Çıktı Köşe noktaları kümesi $V = \{0, 1, \dots, n - 1\}$ ve kökü “0” olan köşe noktaları adlandırılmış köklü ağaç $T = (V, E)$.

Adım 1 $V \leftarrow \{0, 1, 2, \dots, n-1\}$, $E \leftarrow \emptyset$, $S \leftarrow \{0\}$.

Adım 2 1 ile $n-2$ arasındaki her i tamsayısı için

başla

$W \leftarrow V \setminus (S \cup \{v_i, v_{i+1}, \dots, v_{n-2}\})$

$u_i \leftarrow \min(W)$

$S \leftarrow S \cup \{u_i\}$

$E \leftarrow E \cup \{\{u_i, v_i\}\}$

bitir

Adım 3 $w \leftarrow V \setminus S$, $E \leftarrow E \cup \{\{0, w\}\}$

Adım 4 çıktı $T = (V, E)$

Örnek 2.9. Prüfer dizisi $(2, 4, 3, 2)$ olarak verilen ağaç Algoritma 2.8 yardımıyla aşağıdaki şekilde elde edilebilir.

Öncelikle $W \leftarrow \{\{0, 1, 2, 3, 4, 5\} \setminus \{0\} \cup \{2, 3, 4\}\} = \{1, 5\}$ elde edilir. $\min(W) = 1$ olduğundan $S \leftarrow S \cup \{1\} = \{0, 1\}$ ve $E \leftarrow E \cup \{1, 2\} = \{1, 2\}$ elde edilir. $w \leftarrow V \setminus S = \{2, 3, 4, 5\}$ ve $E \leftarrow E \cup \{\{0, w\}\}$ olur. Ardından Adım 1'e geri dönülür. Tüm iterasyonlar Şekil 2.16, Şekil 2.17 ve Şekil 2.18 ile gösterilmektedir.

Prüfer dizileri Cayley teoreminin ispatında oldukça kullanışlı olmakla birlikte ağaçları kodlamak için de oldukça verimli bir yoldur.

Prüfer dizisinin tanımından köşe noktaları adlandırılan ağaçların yaprakları Prüfer dizisinde geçmez. Dahası, köşe noktalarının etiketleri dizide derecelerinin 1 eksiği kere geçer. Böylece Teorem 2.10 sonuç olarak verilebilir.

Teorem 2.10. n köşe noktalı, $i = 1, 2, \dots, n$ için köşe noktalarının dereceleri d_i olan, köşe noktaları adlandırılmış ağaçların sayısı

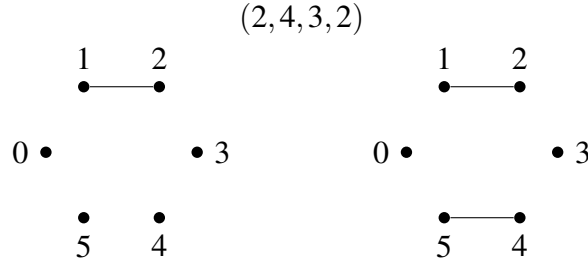
$$\binom{n-2}{d_1-1, d_2-1, \dots, d_n-1} = \frac{(n-2)!}{(d_1-1)!(d_2-1)! \cdots (d_n-1)!}$$

olur.

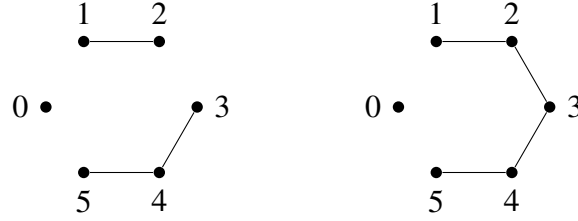
Prüfer dizilerinin yanı sıra derece dizileri bilinen köşe noktaları adlandırılmış ağaçların sayıları Teorem 2.11'de verildiği gibi bulunabilir.

Teorem 2.11. [4] n ($n \geq 3$) köşe noktalı derece dizisi $(d_1, d_2, \dots, d_n)$ olan köşe noktaları adlandırılmış ağaçların sayısı

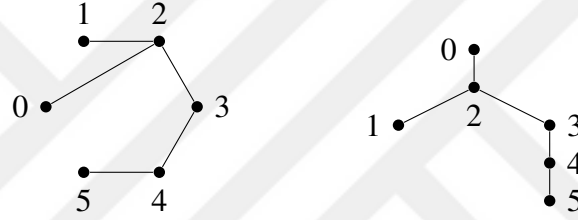
$$\binom{n-2}{d_1-1, d_2-1, \dots, d_n-1}$$



Şekil 2.16. İterasyon 1 ve 2



Şekil 2.17. İterasyon 3 ve 4



Şekil 2.18. İterasyon 5 ve oluşan ağacın farklı bir gösterimi

olur. Burada,

$$\binom{n-2}{d_1-1, d_2-1, \dots, d_n-1} = \frac{(n-2)!}{(d_1-1)!(d_2-1)! \cdots (d_n-1)!}$$

şeklinde çok terimli katsayıyı ifade etmektedir.

Bu Teorem köşe noktaları adlandırılmış ağaçları saymak için oldukça yararlı gibi görünse de izomorfik olmayan ve aynı derece dizilerine sahip ağaçlar nedeniyle beklendiği kadar kullanışlı değildir (Bkz. Şekil 2.19).



Şekil 2.19. İzomorfik olmayan ve aynı derece dizilerine sahip iki ağaç

Genel olarak köşe noktaları adlandırılmış ağaçların sayılarının hesabına kıyasla ele alması çok daha zor olan n köşe noktalı, köşe noktaları adlandırılmamış ağaçların sayısı T_n ile ifade edilir. Köşe noktaları adlandırılmamış ağaçların Cayley Teoremi gibi basit ve bilinen bir formülü yoktur. Ancak George Pólya'nın [15] çalışmasında elde ettiği sonuçlara göre n köşe noktalı köşe noktaları adlandırılmamış ağaçların sayısı asimptotik olarak $an^{\frac{-5}{2}}b^n$ olur. Burada $a = 0.5349 \dots$ ve $b = 2.9557 \dots$ reel sayıdır.

2.2. Bazı Özel Ağaçların Sayıları

Tanım 2.12. İki köşe noktasının derecesi 1 ve kalan tüm köşe noktalarının derecesi 2 olan n noktalı bağlantılı çizgeye patika çizge denir ve P_n ile gösterilir.



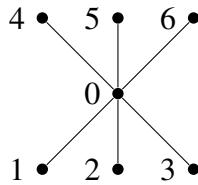
Şekil 2.20. Kökü "0" olan ve köşe noktaları adlandırılmış P_4 patika çizge

Şekil 2.20 ile verilen patika çizgenin Prüfer dizisi $(2, 1)$ olur. Daha genel olarak bir P_n patika çizge için $\{0, 1, \dots, n-1\}$ kümesinin elemanları birer kere kullanılarak oluşturulacak $n-2$ uzunluğundaki her Prüfer dizisi bir patika çizgeye karşılık geleceğinden köşe noktaları adlandırılmış patika çizge sayısı

$$\binom{n}{n-2} (n-2)! = \frac{n!}{2}$$

olur.

Tanım 2.13. Bir köşe noktasının derecesi $n-1$ ve geri kalan tüm noktalarının derecesi 1 olan n köşe noktalı çizgeye yıldız çizge denir ve S_n ile gösterilir.



Şekil 2.21. Kökü "0" olan ve köşe noktaları adlandırılmış yıldız çizge

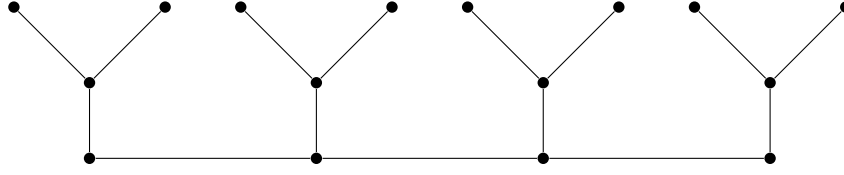
Şekil 2.21 ile verilen yıldız çizgenin Prüfer dizisi $(0, 0, 0, 0, 0)$ olur. Daha genel olarak n köşe noktalı, köşe noktaları $\{0, 1, 2, \dots, n-1\}$ kümesinin elemanları ile adlandırılmış bir S_n yıldız çizgenin derecesi $n-1$ olan köşe noktası $k \in \{0, 1, 2, \dots, n-1\}$ olmak üzere Prüfer dizisi $(k, k, \dots, k)$ şeklinde olur. Dolayısıyla S_n köşe noktaları adlandırılmış yıldız çizge sayısı

$$\binom{n}{1} = n$$

bulunur.

Tanım 2.14. n tane S_k yıldız çizgenin birer yaprağının birbirine bağlanmasıyla elde edilen çizgeye havai fişek çizge (firecracker graph) denir ve $F_{n,k}$ ile gösterilir.

Örneğin Şekil 2.22 ile $F_{4,4}$ havai fişek çizge verilmektedir.



Şekil 2.22. $F_{4,4}$ havai fişek çizge

Tanımdan, $F_{n,k}$ havai fişek çizgenin $n(k-2)$ tane köşe noktasının derecesi 1, 2 tane köşe noktasının derecesi 2, $n-2$ tane köşe noktasının derecesi 3 ve n tane köşe noktasının derecesi $k-1$ olmak üzere toplam nk köşe noktası vardır. Maalesef bu ağaçların sayısı için Prüfer dizileriyle kolayca belirlenebilecek bir yol yoktur. Ancak aşağıdaki gibi kombinatoriyal yöntemlerle $F_{n,k}$ havai fişek çizgelerin sayısı belirlenebilir.

Önerme 2.15. $n \geq 2$ olmak üzere köşe noktaları adlandırılmış $F_{n,k}$ havai fişek çizgelerin sayısı

$$\frac{(nk)!}{2(k-2)!^2}$$

ile bulunur.

Kanıt. $F_{n,k}$ havai fişek çizgenin köşe noktası sayısı nk 'dir. Öncelikle bu noktaların k tanesi ilk yıldız çizge S_k 'dan seçilir. Bu seçim $\binom{nk}{k}$ farklı şekilde yapılır. Daha sonra kalan $nk-k$ köşe noktasından k tanesi ikinci yıldız çizgeden seçilir. Benzer şekilde devam edilirse son adımda kalan $nk-(n-1)k$ köşe noktasından k tanesi kalan son yıldız çizgeden seçilir. Tüm bu seçimler

$$\begin{aligned} & \binom{nk}{k} \binom{nk-k}{k} \dots \binom{nk-(n-1)k}{k} \\ &= \frac{(nk)!}{(nk-k)!k!} \frac{(nk-k)!}{(nk-2k)!k!} \dots \frac{(nk-(n-1)k)!}{(nk-nk)!k!} \\ &= \frac{(nk)!}{k!^n} \end{aligned}$$

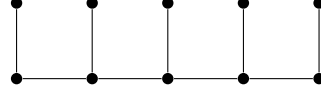
farklı şekilde yapılabilir. Her yıldız çizge S_k , k farklı şekilde etiketlenebileceğinden ve her yıldız çizge için birbirine bağlanacak yapraklar $k-1$ farklı şekilde seçilebileceğinden havai fişek çizgelerin sayısı

$$\frac{(nk)!k^n(k-1)^n}{2k!^n} = \frac{(nk)!}{2(k-2)!^n}$$

elde edilir. $n > 1$ iken yıldız çizgelerin soldan sağa doğru ya da sağdan sola doğru

ele alınması elde edilen havai fişek çizmeyi değiştirmeyeceğinden yukarıdaki ifade 2'ye bölünmüştür. $n = 1$ için 2'ye bölmek gerekmez. $\square$

Tanım 2.16. n -kırkayak ağaç c_n , $2n$ köşe noktalı, n tane P_2 patika çizgenin birer yaprağını birleştirerek oluşturulan ağaçtır (Bkz. Şekil 2.23).



Şekil 2.23. 5-kırkayak ağaç

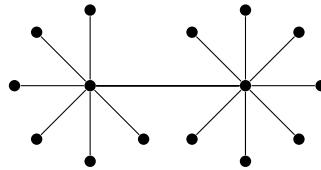
Önerme 2.17. $n \geq 2$ olmak üzere, köşe noktaları adlandırılmış n -kırkayak ağaç sayısı

$$\binom{2n}{n} \frac{(2n-2)!}{2^{(n-2)}}$$

olur.

Kanıt. n -kırkayak ağacın $2n$ köşe noktası vardır ve bunlardan n tanesinin derecesi 1, 2 tanesinin derecesi 2 ve $n-2$ tanesinin derecesi 3 olacağından n -kırkayak çizgenin Prüfer dizisinin uzunluğu $(2n-2)$ 'dir. Dolayısıyla $2n$ köşe noktası arasından n köşe noktası $\binom{2n}{n}$ farklı şekilde seçilir ve seçilen bu köşe noktaları $2n-2$ uzunluğundaki Prüfer dizisine yerleştirilirse seçilen n köşe noktasından $n-2$ tanesi dizide 2 defa geçeceğinden $2^{(n-2)}$ ile bölündüğünde istenilen formül elde edilir. $\square$

Tanım 2.18. S_r ve S_k yıldız çizgelerinin merkezlerinin birbirine bir kenar yardımıyla bağlanmasıyla elde edilen çizmeye çift-yıldız ağaç (double-star tree) denir ve $S_{r,s}$ sembolü ile gösterilir (Bkz. Şekil 2.24).



Şekil 2.24. $S_{7,8}$ çift yıldız ağaç

Tanımdan, merkezdeki köşe noktaları komşu olmak zorundadır. Bu durumda köşe noktalarından bir tanesinin derecesi r , bir tanesinin derecesi s ve geri kalan $r+s-2$ köşe noktasının derecesi 1 olur. Prüfer dizisi de $r+s-2$ uzunluğunda olur. Dolayısıyla her terimi $0, 1, \dots, r+s-1$ kümesinden seçilen 2 elemanın $r+s-2$ uzunluğundaki bir diziye yerleştirilmesiyle oluşturulan her Prüfer dizisi bir çift yıldız çizmeye karşılık gelir. Böylece aşağıdaki sonuç verilebilir.

Önerme 2.19. Köşe noktaları adlandırılmış $S_{r,s}$ çift yıldız ağaç sayısı

$$\binom{r+s}{2} \frac{(r+s-2)!}{(r-1)!(s-1)!} = \frac{(r+s)!}{2(r-1)!(s-1)!}$$

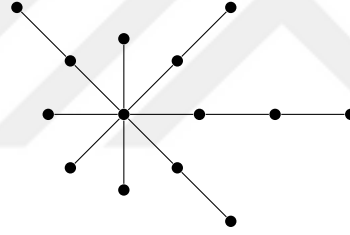
olur.

Kanıt. $S_{r,s}$ çift yıldız ağacın Prüfer dizisi, yaprak olmayan iki noktasının etiketlerine göre değişir ve bu iki köşe noktası $\{0, 1, \dots, r+s-1\}$ kümesindeki $r+s$ elemanın arasından $\binom{r+s}{2}$ farklı şekilde seçilir. Öte yandan oluşturulan $r+s-2$ uzunluğundaki Prüfer dizisinde bu iki köşe noktası

$$\frac{(r+s-2)!}{(r-1)!(s-1)!}$$

farklı şekilde sıralanır çünkü Prüfer dizisinde yaprak olmayan bu köşe noktalarından biri $r-1$ diğeri ise $s-1$ kere geçer. $\square$

Tanım 2.20. Örümcek ağaç (spider tree) veya kısaca örümcek, en fazla 1 köşe noktasının derecesi ikiden büyük olan ağaçtır (Bkz. Şekil 2.25).



Şekil 2.25. 14 köşe noktalı ve 8 yapraklı örümcek ağaç

Eğer bir örümcek ağacın k yaprağı var ise açıktır ki merkezindeki köşe noktasının derecesi k olur. Dolayısıyla, n köşe noktalı k yapraklı bir örümcek çizgenin Prüfer dizisi

$$(v_0, v_0, \dots, v_0, v_1, v_2, \dots, v_{n-k-1})$$

şeklindeki dizinin terimlerinin permütasyonlarından oluşur. $v_0 \in \{0, 1, 2, \dots, n-1\}$ merkezdeki nokta ve $v_i \in \{0, 1, 2, \dots, n-1\} \setminus \{v_0\}$, $i = 1, 2, \dots, n-k-1$. v_0 köşe noktası dizide $k-1$ kere geçer.

Önerme 2.21. $n \geq 2$ olmak üzere köşe noktaları adlandırılmış n köşe noktalı k yapraklı tüm örümceklerin sayısı

$$\binom{n}{n-k} \binom{n-k}{1} \frac{(n-2)!}{(k-1)!}$$

olur.

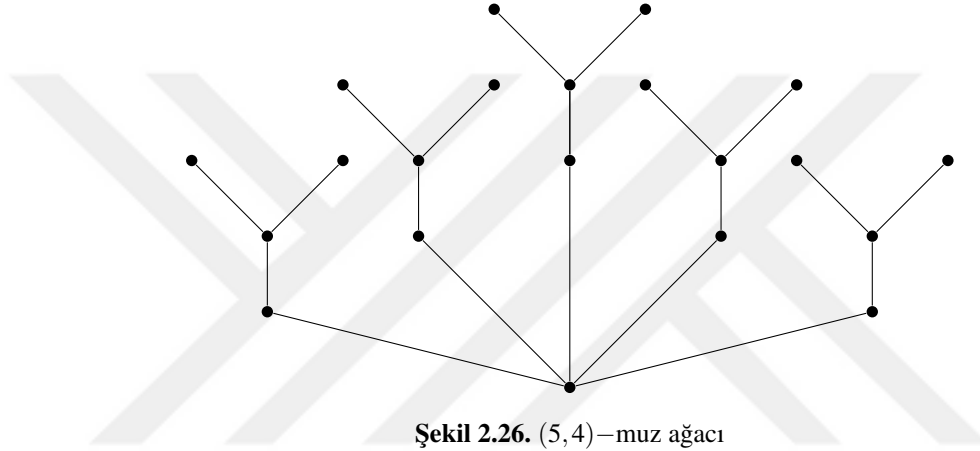
Kanıt. Öncelikle Teorem 2.2 gereği $n \geq 2$ için n köşe noktalı bir ağacın en az iki yaprağı vardır. Örümcek ağacın Prüfer dizisinin uzunluğu $n-2$ ve bu dizi $\{0, 1, 2, \dots, n-1\}$

kümesinin $n - k$ farklı elemanını içerir. Bu elemanlar $\binom{n}{n-k}$ farklı şekilde seçilebilir ve bu $n - k$ köşe noktasından bir tanesi dizide $k - 1$ kere geçtiğinden bu köşe noktası $n - k$ farklı şekilde seçilebilir $k - 1$ kere geçen eleman tekrarlı permütasyondan

$$\frac{(n-2)!}{(k-1)!}$$

olur ve istenilen formül elde edilir. $\square$

Tanım 2.22. n tane S_k yıldız çizgenin birer yaprağının bir tek köşe noktası ile birleştirilmesiyle oluşturulan ağaca (n, k) -muz ağacı (banana tree) denir (Bkz. Şekil 2.26).



Şekil 2.26. $(5, 4)$ -muz ağacı

(n, k) -muz ağacının $nk + 1$ köşe noktası ve nk kenarı vardır. Dolayısıyla, köşe noktaları etiketlenilmiş (n, k) -muz ağacı sayısı Prüfer dizisi yardımıyla bulunabilir.

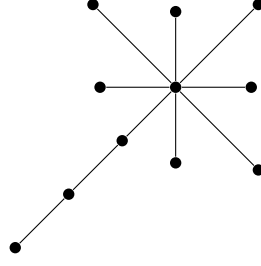
Önerme 2.23. Köşe noktaları etiketlenilmiş (n, k) -muz ağaçlarının sayısı

$$\binom{nk+1}{1} \binom{nk}{n} \binom{nk-n}{n} \frac{(nk-1)!}{(n-1)!((k-1)!)^n}.$$

olur.

Kanıt. (n, k) -muz ağacının derecesi n olan merkezdeki köşe noktası $\binom{nk+1}{1}$ farklı şekilde seçilebilir ve Prüfer dizisinde $n - 1$ kere geçer. Toplamda n tane yıldız olacağından bu yıldızların merkez köşe noktaları ise $\binom{nk}{n}$ farklı şekilde seçilebilir ve her biri Prüfer dizisinde $k - 1$ kere geçer. Her bir yıldız merkez köşe noktasına bağlayan köşe noktaları ise n tanedir ve kalan $nk - n$ köşe noktası arasından $\binom{nk-n}{n}$ farklı şekilde seçilebilir ve Prüfer dizisinde 1 kere geçer. Prüfer dizisinin uzunluğu $nk - 2$ olduğundan önerme ile verilen formül sağlanır. $\square$

Tanım 2.24. n köşe noktalı r -kuyruklu yıldız çizge (r -comet tree of order n) r köşe noktalı bir patika çizgenin bir ucuna $n - r$ ilave kenar eklenerek oluşturulan çizgedir (Bkz. Şekil 2.27).



Şekil 2.27. 11'inci dereceden 4-kuyruklu yıldız çizge

Önerme 2.25. n köşe noktalı köşe noktaları adlandırılmış

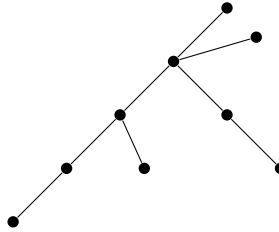
$$\binom{n}{1} \binom{n-1}{r-2} \frac{(n-2)!}{(n-r)!}$$

tane r -kuyruklu yıldız çizge vardır.

Kanıt. n köşe noktalı bir r -kuyruklu yıldız çizgenin Prüfer dizisi $n-2$ uzunluğundadır ve yaprak olmayan $r-1$ köşe noktasının etiketine göre değişir. Bu köşe noktalarından derecesi $n-r+1$ olan köşe noktası n farklı şekilde seçilir. Kalan $r-2$ köşe noktası $\binom{n-1}{r-2}$ farklı şekilde seçilir. $n-2$ uzunluğundaki Prüfer dizisi $(n-2)!$ farklı şekilde sıralanır. Ayrıca Prüfer dizisindeki terimlerden biri dizide $n-r$ kere geçer. Bu durumda sıralama $\frac{(n-2)!}{(n-r)!}$ farklı şekilde yapılabilir. Dolayısıyla n köşe noktalı köşe noktaları adlandırılmış $\binom{n}{1} \binom{n-1}{r-2} \frac{(n-2)!}{(n-r)!}$ tane r -kuyruklu yıldız vardır. $\square$

Tanım 2.26. Tüm yaprakları silindiğinde patika çizge elde edilen, en az 3 köşe noktalı çizgelere tırtıl çizge adı verilir. Elde edilen patika çizgeye ise omurga (spine) adı verilir.

Örneğin, Şekil 2.28 ile 9 köşe noktalı bir tırtıl çizge verilmektedir.



Şekil 2.28. 9 köşe noktalı bir tırtıl çizge

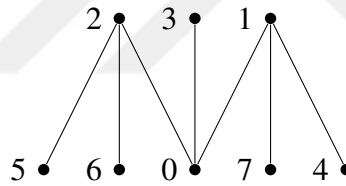
Tanımından da anlaşılacağı üzere, patika çizge, yıldız çizge, çift yıldız çizge ve kuyruklu yıldız çizge birer tırtıl çizgelerdir. Tırtıl çizgenin Prüfer dizisi omurgada kalan köşe noktalarından oluşur.

Köşe noktaları adlandırılmış n köşe noktalı tırtılların sayısı için Prüfer dizisinden yararlanarak basit bir formül vermek mümkün değildir. Ancak köşe noktaları adlandırılmış, n köşe noktalı tırtılların sayısını veren bir formül literatürde yer

$$2^{n-4} + 2^{\lfloor \frac{n}{2} - 2 \rfloor}$$

Tanım 2.27. *Yaprakları hariç geri kalan tüm köşe noktalarının dereceleri sabit bir k tam sayısı olan ağaçlara k -Cayley ağacı denir.*

olarak 4-Cayley ağaçlara Quadtree denir ve çok boyutlu veri tabanlarının inşasında kullanılırlar (örneğin, haritacılık, bilgisayar grafikleri ve görüntü işleme).



Şekil 2.30 ile verilen 3-Cayley ağacının Prüfer dizisi $(0, 1, 2, 2, 0, 1)$ 'dir. Görüldüğü üzere yaprak olmayan köşe noktalarının her biri dizide iki kere geçer. Yani, bir n -Cayley ağacı için yaprak olmayan köşe noktaları Prüfer dizisinde $n - 1$ kere geçer.

$$l = \frac{nk - 2(n-1)}{k-1}$$

Kanıt. n köşe noktalı bir ağacın kenar sayısı $n - 1$ olur. Bu ağacın dereceleri toplamı kenar sayısının iki katını vereceğinden

21

olur. Sadeleştirilecek olursa

$$\begin{aligned} nk - kl + l &= 2(n-1) \\ l(1-k) &= 2(k-1) - kl \\ l &= \frac{2(n-1) - nk}{1-k} \end{aligned}$$

ya da

$$l = \frac{nk - 2(n-1)}{k-1}$$

elde edilir. $\square$

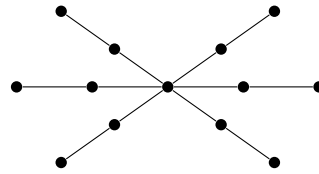
Önerme 2.29. n köşe noktalı köşe noktaları adlandırılmış k -Cayley ağaçlarının sayısı l ağacın yaprak sayısını belirtmek üzere

$$\binom{n}{n-l} \frac{(n-2)!}{((k-1)!)^{n-l}}$$

formülü ile bulunur.

Kanıt. n köşe noktalı bir k -Cayley ağacının l tane yaprağı varsa bu l yaprak Prüfer dizisinde geçmez. Geri kalan $n-l$ köşe noktası ise $k-1$ kere geçer. Bu $n-l$ köşe noktası $\binom{n}{n-l}$ farklı şekilde seçilir. Prüfer dizisinin uzunluğu $n-2$ olacağından bu dizi $(n-2)!$ farklı şekilde sıralanır ve son olarak dizide geçen $n-l$ tane köşe noktasının her biri dizide $k-1$ kere geçtiğinden $((k-1)!)^{n-l}$ ile bölünürse verilen formüle ulaşılır. $\square$

Tanım 2.30. Bir S_k yıldız çizgenin yapraklarına P_{m-1} patika çizgelerin eklenmesiyle elde edilen çizgelere (k, m) -yıldız çizge denir (Bkz. Şekil 2.31).



Şekil 2.31. $(6,2)$ -yıldız çizge

Önerme 2.31. Köşe noktaları adlandırılmış (k, m) -yıldız çizge sayısı

$$\binom{km+1}{k+1} (k+1) \frac{km-1}{(k-1)!}$$

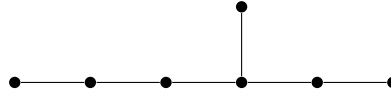
olur.

Kanıt. Verilen ağacın merkezindeki nokta r olmak üzere bir (k, m) -yıldız çizgenin yaprakları hariç tüm köşe noktaları Prüfer dizisinde geçer ve merkezdeki r ile adlandırılan köşe noktası ise dizide $k-1$ kere geçer. Söz konusu ağacın $km+1$ tane köşe noktası

olduğundan dizide geçecek bu $k + 1$ nokta $\binom{km+1}{k+1}$ farklı şekilde seçilir ve merkez köşe noktası ise bu $k + 1$ köşe noktası arasından $k + 1$ farklı şekilde seçilir. Son olarak Prüfer dizisi $km - 1$ uzunluğunda olacağından ve r noktası dizide $k - 1$ kere geçeceğinden $\frac{km-1}{(k-1)!}$ ile çarpılır. Böylece önermede verilen formül elde edilir. $\square$

Tanım 2.32. k ayaklı bir örümcek ağacın bu k ayağının uzunlukları sırasıyla $1, 2, \dots, k$ ise bu tür ağaçlara zeytin ağacı denir ve O_k simgesiyle gösterilir.

Örneğin Şekil 2.32 ile O_3 zeytin ağacı gösterilmektedir.



Şekil 2.32. O_3 zeytin ağacı

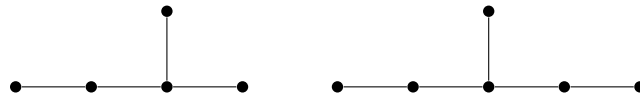
Bir O_k zeytin ağacının köşe noktası sayısı $\frac{k(k+1)}{2} + 1$ olduğundan örümcek ağaç sayısını veren formülde yerine yazılır yaprak sayısına k yazılırsa köşe noktaları adlandırılmış O_k zeytin ağacı sayısı elde edilir. Bu durumda köşe noktaları adlandırılmış O_k zeytin ağaçlarının sayısı

$$\left(\frac{\frac{k(k+1)}{2} + 1}{\frac{k(k+1)}{2} + 1 - k} \right) \binom{\frac{k(k+1)}{2} + 1 - k}{1} \frac{(\frac{k(k+1)}{2} + 1 - 2)!}{(k-1)!}$$

olur.

Tanım 2.33. $(n - 1)$ köşe noktası olan bir P_{n-1} patika çizgenin $\lfloor \frac{n-1}{2} \rfloor$ 'nci köşe noktasına bir kenar eklenerek elde edilen n köşe noktalı ağaca n -yarı-patika (semi-path) denir.

Örneğin Şekil 2.33 ile sırasıyla tek ve çift köşe noktalı iki yarı patika verilmiştir.



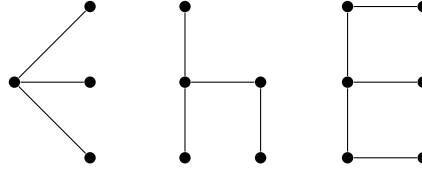
Şekil 2.33. Sırasıyla 5 ve 6 köşe noktalı iki yarı patika çizge

Açıktır ki, n -yarı-patikanın 3 yaprağı vardır. Yeni kenarın eklendiği köşe noktasının derecesi 3 olduğundan Prüfer dizisinde bu nokta 2 kez yer alır. Bu nokta n farklı şekilde seçilebilir. Geriye kalan ve dereceleri 2 olan $n - 4$ köşe noktası ise $\binom{n-1}{n-4}$ farklı şekilde seçilebilir. Bu durumda köşe noktaları adlandırılmış n -yarı-patikaların sayısı

$$\binom{n}{1} \binom{n-1}{n-4} \frac{(n-2)!}{2}$$

olur.

Tanım 2.34. n köşe noktalı ve tam olarak 3 yaprağı bulunan ağaçlara tripod denir.



Şekil 2.34. Üç farklı tripod ağaç

3 yaprağı olması gerektiğinden açıktır ki $n \geq 4$ olur. Tripod ağaçlara Şekil 2.34 ile üç farklı örnek verilmiştir.

Açıktır ki tripod ağaçlarda köşe noktalarından bir tanesinin derecesi 3, 3 tanesinin 1 ve geri kalan $n - 4$ köşe noktasının derecesi ise 2'dir. Dolayısıyla Prüfer dizisinde derecesi 3 olan köşe noktası $\binom{n}{1}$ farklı şekilde seçilebilir ve Prüfer dizisinde 2 kere geçer. Kalan derecesi 2 olan köşe noktaları ise $n - 1$ nokta arasından $\binom{n-1}{n-4}$ farklı şekilde seçilebilir ve Prüfer dizisinde birer kere geçerler. Dolayısıyla n köşe noktalı köşe noktaları adlandırılmış tripod ağaç sayısı

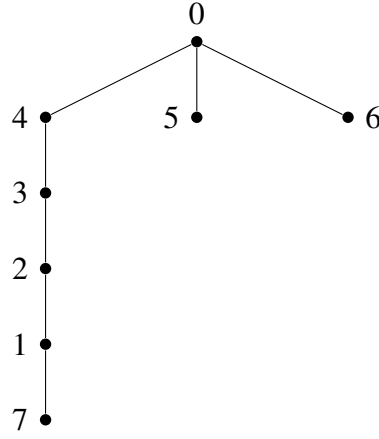
$$\binom{n-1}{n-4} \frac{(n-2)!n}{2}$$

olur.

2.3. Rastgele Ağaçlar Üretmek

n köşe noktalı rastgele ağaç üreten bir prosedür yazılması istensin. Öyle ki her bir ağacın ortaya çıkma olasılığı eşit olsun. Prüfer dizisi sayesinde bu oldukça kolay bir şekilde yapılabilir. Tek yapılması gereken $n - 2$ tane tam sayıyı 0 ile $n - 1$ arasındaki tam sayılardan seçerek oluşturmaktır. Veya alternatif olarak 1 ile n^{n-2} arasından bir sayıyı n tabanına çevirerek bir Prüfer dizisi oluşturulabilir. Eğer n tabanına çevrildiğinde oluşan sayının terim sayısı $n - 2$ 'den küçükse $n - 2$ 'ye tamamlayacak şekilde başına 0 eklenir. Bu ikinci metot sayesinde n köşe noktalı tüm ağaçlar numaralandırılabilir.

Örneğin 8 köşe noktalı 668'inci ağaç için 668, 8 tabanına çevrilirse $(1234)_8$ elde edilir ve terim sayısı 4 olduğundan başına 2 tane 0 eklenerek istenilen Prüfer dizisi $(0, 0, 1, 2, 3, 4)$ olarak elde edilir (Bkz. Şekil 2.35).



Şekil 2.35. Prüfer dizisi (0, 0, 1, 2, 3, 4) olan ağaç

3. AĞAÇLAR İÇİN MAPLE PROSEDÜRLERİ

Bu bölümde ağaçlar için Prüfer dizileri yardımıyla çeşitli Maple prosedürleri verilmektedir. Maple'ın seçilme nedenlerinin başında sembolik hesaplamadaki gücü ve geniş çaplı kullanıcı kitlesine sahip olmasıdır.

Paketteki bazı komutlar ve kısaca işlevleri aşağıdaki gibidir.

BananaTree Bu komut (n, k) —banana tree oluşturur.

Caterpillar Bu komut n köşe noktalı omurgası k köşe noktalı bir tırtıl çizge oluşturur.

Centipede Bu komut c_n kırkayak çizge oluşturur.

Comet Bu komut n köşe noktalı bir r —kuyruklu yıldız çizge oluşturur.

DoubleStar Bu komut $S_{r,s}$ çift yıldız çizge oluşturur.

FireCracker Bu komut (n, k) —havai fişek çizge oluşturur.

GetTree Bu komut n köşe noktalı köşe noktaları adlandırılmış ağacı verir.

Prufer2Tree Prüfer dizisi verilen köşe noktaları adlandırılmış ağacı verir.

RandomTree Rastgele bir ağaç verir.

SemiPath n köşe noktalı yarı-patika çizgeyi verir.

Spider n köşe noktalı k yapraklı örümcek çizgeyi verir.

Tree2Prufer Verilen ağacın Prüfer dizisini verir.

Kaynak kodlarının tamamı ekler bölümünde yer almaktadır.

3.1. Komut Adı: Prufer2Tree

Komut Prufer2Tree(P)

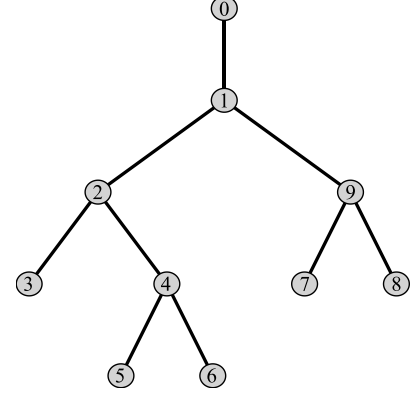
Parametreler P - list

Tanım The Prufer2Tree komutu Prüfer dizisini parametre olarak alır ve bu Prüfer dizisine sahip olan ağacı oluşturur. Burada uygulanan strateji Algoritma 2.8 ile

verilen aloritmadan yararlanmaktadır.

Örnek

```
> P := [2,4,4,2,1,9,9,1];  
> T := Prufer2Tree(P);  
> DrawGraph(T,style=tree);
```



Şekil 3.36. Prufer2Tree komutu ile oluşturulmuş bir ağaç

3.2. Komut Adı: Tree2Prufer

Komut Tree2Prufer(T)

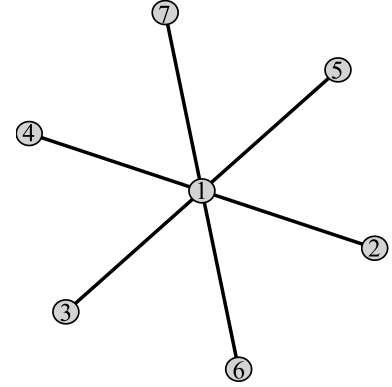
Parametreler T - Graph

Tanım The Tree2Prufer komutu verilen ağacın Prüfer dizisini oluşturur. Uygulanan metot Algoritma 2.5 olarak verilen algoritmayı kullanmaktadır.

Örnek

```
> T := CompleteGraph(1,6);  
> P := Tree2Prufer(T);  
> DrawGraph(T,style=spring);
```

P := [1, 1, 1, 1, 1]



Şekil 3.37. T ağacının Tree2Prufer komutu ile oluşturulmuş kodu

3.3. Komut Adı: RandomTree

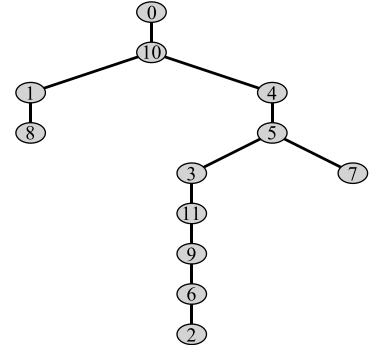
Komut RandomTree(n)

Parametreler n - posint

Tanım RandomTree komutu köşe sayısı verildiğinde rastgele bir ağaç oluturur. Uygulanan metot verilen köşe sayısı ile rastgele bir Prüfer dizisi oluşturup daha sonra bu Prüfer dizisini Prufer2Tree komutu kullanarak bir ağaç elde edilir..

Örnek

```
> T := RandomTree(12);  
> DrawGraph(T,style=tree);
```



Şekil 3.38. RandomTree komutu ile oluşturulmuş rastgele ağaç

3.4. Komut Adı: GetTree

Komut GetTree(n,k)

Parametreler

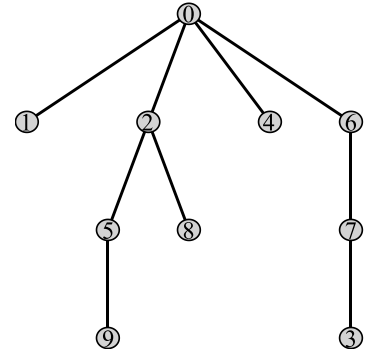
n - posint

k - nonnegint 0 ile $n^{n-2} - 1$ arasındaki tamsayılar.

Description GetTree komutu n köşe noktalı ve köşe noktaları adlandırılmış n^{n-2} ağaçtan k . ağacı verir.

Örnek

```
> T := GetTree(10,12345678);  
> DrawGraph(T,style=tree);
```



Şekil 3.39. GetTree komutu ile oluşturulmuş 10 köşe noktalı 12345678. ağaç

3.5. Komut Adı: DoubleStar

Komut DoubleStar(r,s)

Parametreler

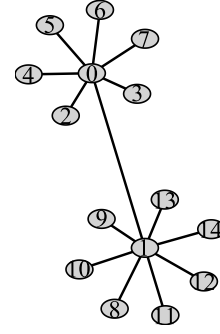
r - posint

s - posint

Tanım DoubleStar komutu $S_{r,s}$ çift-yıldız ağacı verir.

Örnek

```
> T := DoubleStar(6,7);  
> DrawGraph(T,style=spring);
```



Şekil 3.40. DoubleStar komutunun uygulandığı

3.6. Komut Adı: Centipede

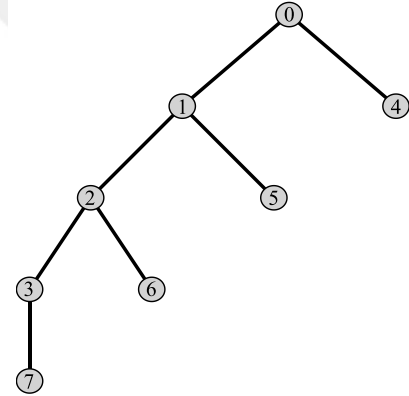
Komut Centipede(n)

Parametreler n - Pozitif tam sayı (patika çizge P_n üzerindeki köşe noktalarının sayısı)

Tanım Centipede komutu c_n kırkayak çizgeyi verir.

Örnek

```
> T := Centipede(4);  
> DrawGraph(T,style=tree);
```



Şekil 3.41. Centipede komutunun uygulandığı

3.7. Komut Adı: FireCracker

Komut FireCracker(n,k)

Parametreler

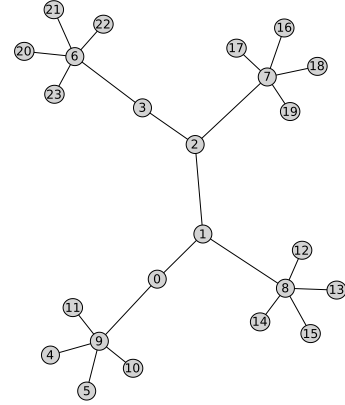
n - posint

k - posint

Tanım FireCracker komutu $F_{n,k}$ havai fişek çizgeyi verir.

Örnek

```
> T := FireCracker(4,6);  
> DrawGraph(T,style=spring);
```



Şekil 3.42. FireCracker komutunun uygulanişı

3.8. Komut Adı: BananaTree

Komut BananaTree(n,k)

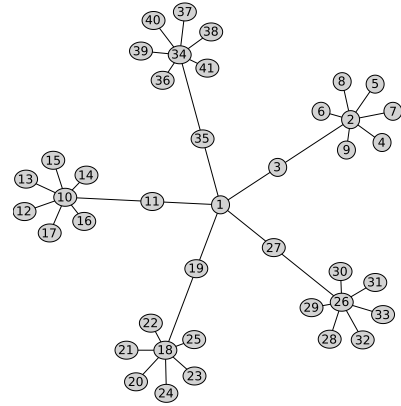
Parametreler

n - posint
 k - posint

Tanım BananaTree komutu (n,k) -banana tree oluşturur.

Örnek

```
> T := BananaTree(5,7);  
> DrawGraph(T,style=spring);
```



Şekil 3.43. BananaTree komutunun uygulanişı

3.9. Komut Adı: Caterpillar

Komut Caterpillar(n,k)

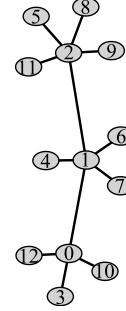
Parametreler

n - posint
 k - posint

Tanım Caterpillar komutu omurgası k köşe noktalı olan n köşe noktalı rastgele bir tırtıl çizge oluşturur.

Örnek

```
> T := Caterpillar(13,3);
> DrawGraph(T,style=spring);
```



Şekil 3.44. Caterpillar komutunun uygulanişı

3.10. Komut Adı: SemiPath

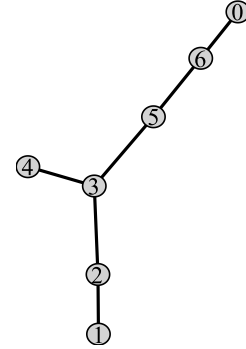
Komut SemiPath(n)

Parametreler n - posint

Tanım SemiPath komutu n yarı-patikayı oluşturur.

Örnek

```
> T := SemiPath(7);
> DrawGraph(T,style=spring);
```



Şekil 3.45. SemiPath komutunun uygulanişı

3.11. Komut Adı: Comet

Komut Comet(n, r)

Parametreler

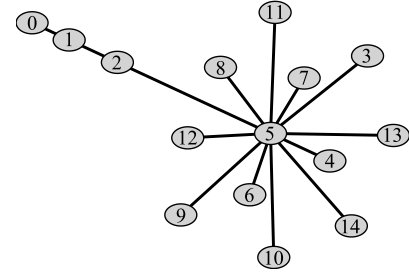
n - posint

r - posint

Tanım Comet komutu n köşe noktalı bir r -kuyruklu yıldız oluşturur.

Örnek

```
> T := Comet(15,4);  
> DrawGraph(T,style=spring);
```



Şekil 3.46. Comet komutunun uygulanışı

3.12. Komut Adı: Spider

Komut Spider(n,k)

Parametreler

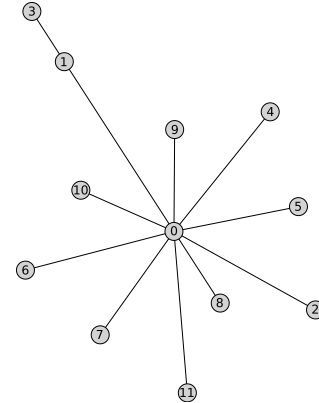
n - posint

k - posint

Tanım Spider komutu n köşe noktalı k yapraklı bir örümcek oluşturur.

Örnek

```
> T := Spider(10,12);  
> DrawGraph(T,style=spring);
```



Şekil 3.47. Spider komutunun uygulanışı

3.13. Komut Adı: CayleyTree

Komut CayleyTree(n,r)

Parametreler

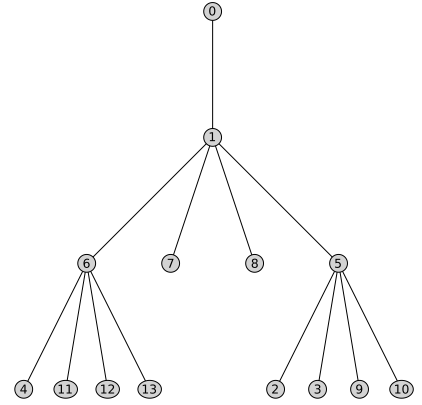
n - posint

r - posint

Tanım CayleyTree komutu n köşe noktalı bir r -Cayley ağacı oluşturur.

Örnek

```
> T := CayleyTree(14,15);  
> DrawGraph(T);
```



Şekil 3.48. CayleyTree komutunun uygulanışı

3.14. Komut Adı: Tripod

Komut Tripod(*n*)

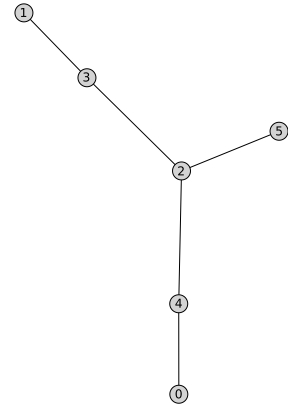
Parametreler

n - posint

Tanım Tripod komutu *n* köşe noktalı bir Tripod ağaç oluşturur.

Örnek

```
> T := Tripod(6);  
> DrawGraph(T,style=spring);
```



Şekil 3.49. Tripod komutunun uygulanışı

EK MAPLE KODLARI

3. kısımda verilen MAPLE kodlarının rutinleri aşağıdadır.

```
with(GraphTheory):
Prufer2Tree := proc(P::list)::Graph:
    local i,n,u,e,V,T,E,S,W;
    n := nops(P) + 2;
    V := {seq(i,i=0..n-1)};
    T := Graph(convert(V,list));
    S := {0};
    for i from 1 to n-2 do
        W := V minus (S union {op(i..n-2,P)} );
        u := min(W);
        S := S union {u};
        e := {P[i],u};
        AddEdge(T,e);
    od:
    W := V minus S;
    e := {W[1],0};
    AddEdge(T,e);
    return T;
end proc:

Tree2Prufer := proc(T::Graph)::list;
    local G,V,L,i;
    if not IsTree(T) then
        error "Argument is not a tree!";
    end if;
    L := [];
    G := T;
    V:=sort(Vertices(G));
    while nops(V)>2 do
        for i from 1 to nops(V) do
            if Degree(G,V[i])=1 and V[i]<>0 then
                L := [op(L), op(Neighbors(G,V[i])) ];
                G := DeleteVertex(G,V[i]);
                break;
            end if:
        od:
        V:=sort(Vertices(G));
    end while;
```

```

    od:
    return L;
end proc:

RandomTree := proc(n::posint)::Graph;
    local P;
    P := RandomTools[Generate]
        (list(integer(range=0..n-1),n-2));
    return Prufer2Tree(P);
end proc:

GetTree := proc(n::posint,k::nonnegint)::Graph;
    local L,i,m;
    if k>=n^(n-2) then
        error
            "second argument must be smaller than %1",n^(n-2);
    end if:
    L := convert(k,base,n);
    while ListTools[SelectFirst](L)=0 do
        L := [ op(2..nops(L),L) ];
    od:
    for i from 1 to n-nops(L)-2 do
        L:=[0,op(L)];
    od:
    return Prufer2Tree(L);
end proc:

DoubleStar := proc(r::posint, s::posint)::Graph;
    local P;
    P := [seq(0,i=1..r),seq(1,i=1..s)];
    return Prufer2Tree(P);
end proc:

Centipede := proc(n::posint)::Graph;
    local P;
    P := [seq(i,i=0..n-2),seq(n-i,i=1..n-1)];
    return Prufer2Tree(P);
end proc:

FireCracker := proc(n::posint,k::posint)::Graph;
    local P,i;

```

```

P := [];
for i from 0 to n-1 do
  P:= [op(P), seq(2*n-i+1,j=1..k-2),i];
od;
P := [op(P),seq(i,i=n-2..1,-1)];
print(P);
return Prufer2Tree(P);
end proc:

BananaTree := proc(n::posint, k::posint)::Graph;
  local P,i,j;
  P:=[];
  for i from 1 to n do
    P:=[op(P),seq(i,j=1..k-1)];
    if i<>n then P := [op(P),0]; end if:
  od:
  return Prufer2Tree(P);
end proc:

Caterpillar := proc(n::posint,k::posint)::Graph;
  local L1,L2,i;
  L1 := combinat[randperm]([seq(i,i=0..k-1)]);
  L2 := RandomTools[Generate](list(integer(range=0..k-1),n-k-2));
  return Prufer2Tree([op(L2),op(L1)]);
end proc:

Spider := proc(n::posint,k::posint)::Graph;
  local P;
  if n<k then
    error "Number of leaves must be smaller than number of vertices";
  end if;
  P := combinat[randperm]([seq(i,i=1..n-k-1),seq(0,i=1..k-1)]);
  return Prufer2Tree(P);
end proc:

SemiPath :=proc(n::posint)::Graph;
  local P,k;
  k := floor(n/2);
  P := [seq(i,i=2..n-1)];
  P[k]:=k;

```

```

    return Prufer2Tree(P);
end proc:

Comet := proc(n::posint,r::posint)::Graph;
    local P;
    P:=[seq(r+1,i=1..n-r),seq(i,i=r-2..1,-1)];
    return Prufer2Tree(P);
end proc:

Tripod := proc(n::posint)::Graph;
    local P,L,i,j;
    if n<4 then
        error "Number of vertices must be greater than 3";
    end if:
    i:= RandomTools[Generate](integer(range=0..n-1));
    L:=[];
    for j from 0 to n-1 do
        if j<>i then L:=[op(L),j]; end if:
    end do:
    L:=combinat[randperm](L);
    P:=combinat[randperm]([i,i,op(1..n-4,L)]);
    return Prufer2Tree(P);
end proc:

CayleyTree := proc(n::posint, k::posint)::Graph;
    local P, K, L, A, A1, l, i;
    if n <= k then error
        "second argument must be smaller than %1",n;
    end if;
    l:=(k*n-2*n+2)/(k-1);
    if evalb(floor((k*n-2*n+2)/(k-1))=l)
    then
        L := [seq(i, i = 0 .. n-1)];
        A := convert(Shuffle(Array(L)), list);
        K := SelectFirst(n-l,A);
        P := [seq(op(K), 1 .. k-1)];
        A1 := Shuffle(Array(P));
        P := convert(A1, list);
    else error "There is no such Cayley Tree!"; end if;
    return Prufer2Tree(P);
end proc;

```

KAYNAKÇA

- [1] KIRCHHOFF, G., Über die Auflösung der Gleichungen, auf welche man bei der Untersuchung der linearen Verteilung galvanischer Ströme geführt wird. *Ann. Phys. Chem.*, 72 (1847) 497-508.
- [2] CAYLEY, A., On the theory of analytical forms called trees. *Philos. Mag.*, 13 (1857) 19-30.
- [3] PRÜFER, H., Neuer Beweis eines Satzes über Permutationen. *Arch. Math. Phys.* **27**, 742–744 (1918).
- [4] MOON, J.W., Counting labelled trees (Canadian mathematical monographs). In *Canadian Mathematical Congress*, 1970.
- [5] CARNIA, E.; SYLVIANI, S., The use of GAP and MAPLE software in teaching group theory, 2017.
- [6] MONAGAN, M., Introduction to Maple's Graph Theory Package, 2013.
- [7] MAPLESOFT, a division of Waterloo Maple Inc.. (2019). Maple. Waterloo, Ontario.
- [8] AKYAR, E., *Çizge Kuramına (Graf Teorisine) Giriş*. Seçkin Yayıncılık, Turkey, 2021.
- [9] CHARTRAND, G.; ZHANG, P., *A First Course in Graph Theory*. Dover Publications, Boston, 2012.
- [10] LOVASZ, L.; PELIKAN, J.; VESTERGOMBI, K.L., *Discrete Mathematics*. Springer-Verlag, Berlin, Heidelberg, 2003.
- [11] WEST, D.B., *Introduction to Graph Theory*. second edn. Pearson, 2000.
- [12] CAYLEY, A., A Theorem on Trees. *Q. J. Math.* **23**, 376–378 (1889).
- [13] WANG, X.; WANG, L. AND WU, Y., An Optimal Algorithm for Prüfer Codes. *Journal of Software Engineering and Applications*, Vol. 2 No. 2, 2009, 111–115.
- [14] CHEN, H.C.; WANG, Y.L., An efficient algorithm for generating Prüfer codes from labelled trees. *Theory of Computing Systems* **33**(1), 97–105 (2000).
- [15] PÓLYA, G., Kombinatorische Anzahlbestimmungen für Gruppen, Graphen und chemische Verbindungen. *Acta Mathematica*, Vol. 68, 1937, 145–254.
- [16] HARRARY, F.; SCHWENK, A.J., The number of caterpillars. *Disc. Math.* **6**(4), 359–365 (1973).

ÖZGEÇMİŞ

ORCID ID : 0000-0001-8216-820X

Adı-Soyadı : Emre Yivli

Yabancı Dil : İngilizce

Eğitim ve Mesleki Geçmişi:

- 2016–2020, Eskişehir Teknik Üniversitesi, Fen Fakültesi, Matematik Bölümü.

