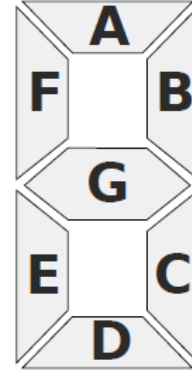


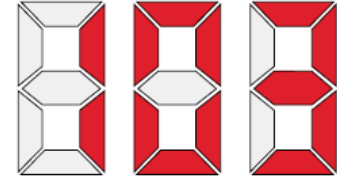
## Dijital Sayı

Kunduz Kaya, ışık yayan diyotları (LED'ler) kullanarak sayıları görüntülemek istiyor. Her sayıyı temsil etmek için 7 LED parçası kullanabiliyor. Parçalar, yanda gösterildiği gibi A, B, C, D, E, F ve G olarak etiketlenmiştir.



Belirli bir LED parçasını açmak için, tabloda karşılık gelen hücreyi belirtmesi gerekir. Örneğin, aşağıdaki tabloda üç basamaklı sayı 103 gösterilecektir:

	A	B	C	D	E	F	G
↓		■	■				
	■	■	■	■	■	■	
	■	■	■	■			■



### Soru

Aşağıdaki tabloyu kullanırsak, ekranda hangi sayı görüntülenir?

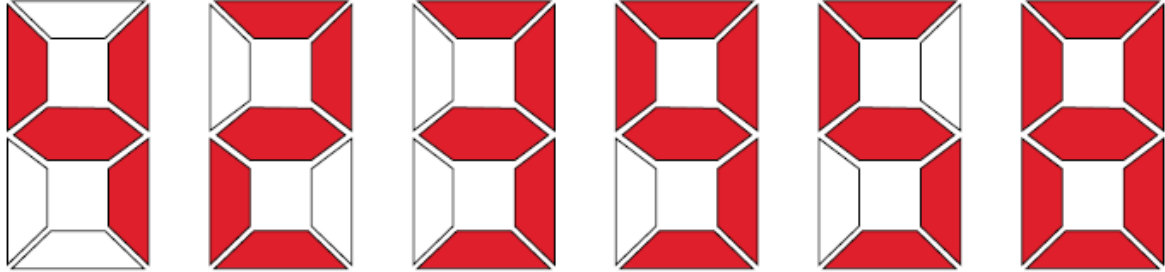
	A	B	C	D	E	F	G
↓		■	■			■	■
	■	■		■	■		■
	■	■	■	■			■
	■	■	■	■		■	■
	■		■	■		■	■
	■	■	■	■	■	■	■

- A) 423958
- B) 624785
- C) 433968
- D) 624958

## Doğru Yanıt

A

## Sorunun Çözümü



Örnekten, tablodaki her satırın bir dijital sayıyı temsil ettiğini fark ettik. Sorunun tablosundaki ilk satır için, yalnızca 4 sayısını temsil eden B, C, F ve G parçaları seçilir. İkinci satır için 2 sayısını temsil eden A, B, D, E, G parçaları seçilir. Her işlem için bu işleme devam edersek, 423958 sayısını buluruz.

## Sorudaki Enformatik Kavramı

Bir nesnenin bir dizi başka nesne tarafından temsil edilmesinin değişimine kodlama denir. Kodlama bilgisayar biliminde çok yaygın kullanılır. Her basamağı, iki olası değerden birine sahip olabilecek yedi hücre dizisi ile değiştiririz: kırmızı veya beyaz. Bu gibi görüntüler bu modern çağda yaygın olarak kullanılmaktadır. Küçük boyuttaki ışık yayan diyotlar (LED'ler) nedeniyle, bazıları 7 parçalı bir ekrana bağlanabilir. 7 parçalı ekranlar kullandığımızda, ABCDEFG için sıfır olanı bir parçayı kapatıp açtığı bir değer olan (kırmızı) veya sıfır (beyaz) olan 8 değer dizisi gönderebiliriz. Örneğin, 7 parçalı bir ekranda "4" sayısı 0011 0011 olarak ifade edilir.

## Anahtar Kelimeler







etkileşimler, sistemler, kodlama

## İlgili Web Siteleri



















<https://tr.wikipedia.org/wiki/LED>

## Doğum Günü Kutlaması

Bir arkadaşları, Sibel ve Levent'i öğleden sonra saat 15:00'de başlayacak olan doğum günü kutlamasına davet etti. Onlar da giderken taze yapılmış kurabiye, kek ve börek götürmeye karar verdiler. Hamur işleri tarifi kitabından aşağıdaki pişirme sürelerini öğrendiler.

			
İşlemler	Börek	Kurabiye	Kek
 Hamuru Hazırlama	45 dk.	30 dk.	15 dk.
 Fırında Pişirme	15 dk.	45 dk.	30 dk.
 Süsleme	0 dk.	15 dk.	30 dk.

Hazırlık sürecini 3 aşamaya böldüler. Levent hamuru hazırlayacak ve sonra fırına verecek. Kek piştikten sonra Sibel krema ile süsleme yapacak. Fırında aynı anda sadece tek bir yiyecek için yer var. Sibel ve Levent aynı anda yalnızca tek bir yiyeceği hazırlayabilir. Saat 13:00 olunca işe başlayıp saat 15:00'de kutlamaya gitmek istiyorlar. O yüzden aşağıdaki planlamayı yaparlar:

	13:00	14:00	15:00	16:00
 Börek	   			
 Kurabiye		     		
 Kek				   

Pişirme sürelerinin uzunluğu dikkatlerini çeker ve saat 15:00'e nasıl yetişeceklerini düşünürler. Bu yüzden zamanında hazır olabilmek için pişirme sırasını değiştirmeye karar verirler.

### Soru

Sibel ve Levent bu 3 hamur işini en erken saat kaçta hazırlayabilir?

- A) 14:15
- B) 14:30
- C) 14:45
- D) 15:00

## Dođru Yanıt

C

### Sorunun Çözümü

Eđer öncelikle kek, sonra kurabiye ve son olarak börek pişirilirse, tüm hamur işleri saat 14:45 itibari ile hazır olabilir. Sibel ve Levent zamanında bitirebilmek için sıralamayı bu şekilde deđiştirir. Böylece 1,5 saat (14:30'a kadar) sıradan bađımsız olarak hamurların hazırlanmasına harcanır. Eđer kek ya da kurabiye sona kalırsa, o zaman pişirme süresi bir saat daha uzamaktadır. Bu nedenle pişirme süresi en kısa olan börek son olarak pişirilir.

### Sorudaki Enformatik Kavramı

Zaman planında görüldüğü üzere, bir hamur işini hazırlama işlemi, 3 alt işleme bölünmüştür: Hamuru Hazırlama, Fırında Pişirme ve Süsleme. Böylece birini bitirip diđerine başlamak yerine eş zamanlı olarak işlemler yürütülebilmektedir. Böylece Sibel kek hamurunu hazırlarken, fırında kurabiyeler pişebilir, ya da Sibel kek hamurunu hazırlarken Levent de kurabiye hamurunu hazırlayabilir. Bu süreç "boru hattı (pipelining)" olarak adlandırılır ve genellikle mikroişlemcilerde kullanılır. Boru hattı yaklaşımı, yönergelerin paralel işlenmesini ve bir işlemcideki kaynakların daha verimli kullanılmasını sağlar.

### Anahtar Kelimeler

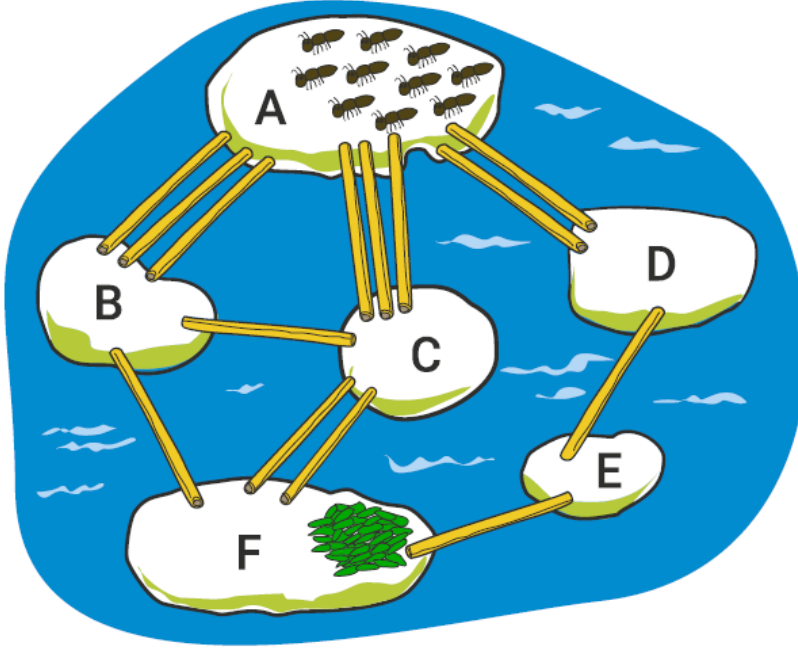
alt işlemlere ayırma, boru hattı, paralel işleme

### İlgili Web Siteleri

[https://tr.wikipedia.org/wiki/Boru\\_hatt%C4%B1\\_\(bilgisayar\)](https://tr.wikipedia.org/wiki/Boru_hatt%C4%B1_(bilgisayar))

## Nehirdeki Karıncalar

On karınca, A taşında bulunur ve F taşındaki yiyeceklere ulaşmaya çalışır. Aynı anda sadece bir karınca tek bir pipet üzerinde yürüyebilir ve bir karıncanın bir taştan diğerine yürümesi 1 dakika sürer.



### Soru

3 dakika sonra F taşı üzerindeki yiyeceğe ulaşabilecek maksimum karınca sayısı kaçtır?

- A) 4
- B) 5
- C) 6
- D) 7

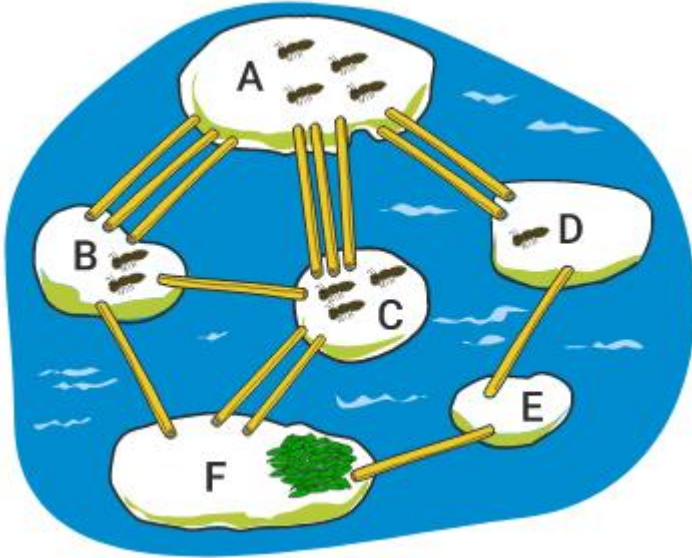
## Doğru Yanıt

D

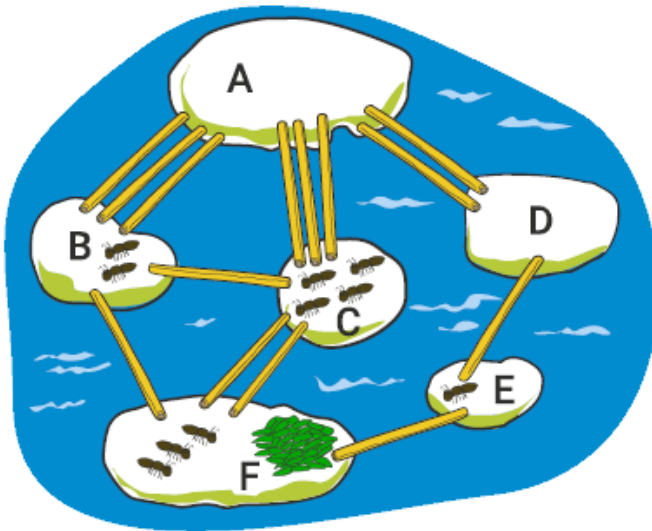
## Sorunun Çözümü

Aşağıdaki görsellerde her bir dakika sonrasındaki olası durum gözlenmektedir:

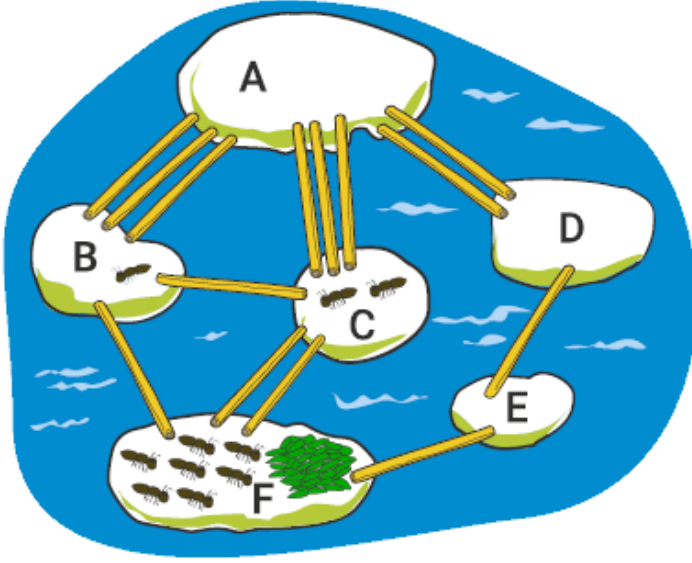
1 dakika sonra:



2 dakika sonra: 3 karınca yiyeceğe ulaşır.



3 dakika sonra: 7 karınca yiyeceğe ulaşır.



### Sorudaki Enformatik Kavramı

Buradaki amaç, karıncaların ağ üzerinden akışını optimize etmektir, böylece yiyeceğe mümkün olduğunca çok sayıda karınca 3 dakika içinde ulaşır. Buna optimizasyon problemi denir. Seyahat ettikleri ağın yapısının farkında olmayan karıncalar en iyi çözümü bulamayacaklardır. Ancak, ağların yapısını görebilen bir gözlemci en uygun stratejiyi bulabilir. Bu görevde, karıncaların ağ yapısının farkında olduklarını ve belirli bir yol boyunca hareket etmeyi seçtiklerini varsayıyoruz.

Grafikler ağları modellemek için kullanılan soyut veri yapılarıdır ve akışı belirli koşullar altında optimize etmek için birçok algoritma vardır. Alternatif olarak önceden var olan bir algoritmadan faydalanmadan şunu söyleyebiliriz:

- D-E taşları ile birden fazla karınca göndermenin bir anlamı yoktur.
- A-B taşları ile ikiden fazla karınca göndermenin bir anlamı yoktur.
- B-C arasındaki pipet akıma katkıda bulunmaz ve göz ardı edilebilir.
- Akışın sınırlayıcı faktörü, B-F ve C-F pipetleridir.

Bu soru bu gerçekler ışığında çözülür.

### Anahtar Kelimeler

optimizasyon problemi, ağ akışı

### İlgili Web Siteleri

[https://en.wikipedia.org/wiki/Flow\\_network](https://en.wikipedia.org/wiki/Flow_network)

[https://en.wikipedia.org/wiki/Maximum\\_flow\\_problem](https://en.wikipedia.org/wiki/Maximum_flow_problem)

[https://en.wikipedia.org/wiki/Optimization\\_problem](https://en.wikipedia.org/wiki/Optimization_problem)

## Yaşlı Kunduzların Mesajı

Bilge Kunduz, barajın dibinde eski bir ağaç parçası keşfetti. Yakından baktığında tahtaya oyulmuş gizemli işaretleri fark eder. Bu mesajın kunduzların barajda yaşadıkları zamandan kaldığını ve bir kodlama tablosu olduğunu düşünür.

	I	II	III	III	○	○	⊖	⊖	○
☀	A	B	C	D	E	F	G	H	I
☞	J	K	L	M	N	O	P	Q	R
☞	S	T	U	V	W	X	Y	Z	

Tabloyu uzun süre inceleyen Bilge Kunduz, sonunda nasıl çalıştığını bulur. Yeni işaretler, ilgili satırlara ve sütunlara atanan sembollerin birleşimi ile oluşmaktadır. Örneğin: <H> harfi aşağıdaki şekilde kodlanmıştır:

	I	II	III	III	○	○	⊖	⊖	○
☀	A	B	C	D	E	F	G	H	I
☞	J	K	L	M	N	O	P	Q	R
☞	S	T	U	V	W	X	Y	Z	

☀ + ⊖ = ☀⊖

Bilge Kunduz barajın farklı yerlerinde bu işaretleri gördüğünü hatırlar. Oraya gider ve gerçekten bir ağaçta aşağıdaki mesajı bulur:



### Soru

Bu bilgilere göre, yaşlı kunduzların mesajı nedir?

- A) LOVEWATER
- B) SLEEPDAYS
- C) LOVEMYSUN
- D) CAREFORME



## Doğru Yanıt

A) LOVEWATER

## Sorunun Çözümü

İlk başta, mesajın uzunluğunun ipucu verip vermediğini kontrol edelim. Ancak tüm çözümler aynı uzunlukta olduğu için bu durum yardımcı olmuyor. O zaman, ilk harfin kodunu çözüp, bunun bir "L" olduğunu buluruz. Bu durum yanıtın B) ya da D) olamayacağını belirtir. Sonra A) ve C) arasındaki farklara bakılır. Beşinci harfte farklılaşma başlıyor. Bu yüzden beşinci işaretin kodunu çözer ve bunun "W" olduğunu buluruz. Bu nedenle çözümün A olduğunu görürüz. Çözümde emin olmak için başka bir işareti, örneğin son işaretin kodunu çözebiliriz. Sonuç "R" ve beklenen çözüme uyuyor.

## Sorudaki Enformatik Kavramı

Veri güvenliği bugün toplumlarda büyük bir sorundur. Verileri yetkili olmayan kişilere karşı koruma yöntemlerinden biri gizli şifrelemedir. Kriptoloji yaklaşık 3500 yıl önce başladı ve ilk yöntem her bir harfi başka bir harfle değiştirmektir. Bu örnekte, bilinen alfabenin harfleri için, şifreleme yöntemini kolayca hatırlayabilecek şekilde yeni işaretler oluşturulur.

Bu görevde, kodlamayı tanımlayacak bir tablo olmasaydı ve bize yalnızca semboller verilseydi, resimlerin kelimelere dönüştürülmesi çok daha zor olurdu. Kodları kırmaya çalışan kriptanalistler, kodlanmış sembollerin ne olması gerektiğini belirlemek için frekans analizi ve örüntü tanıma gibi teknikleri kullanırlar.

## Anahtar Kelimeler

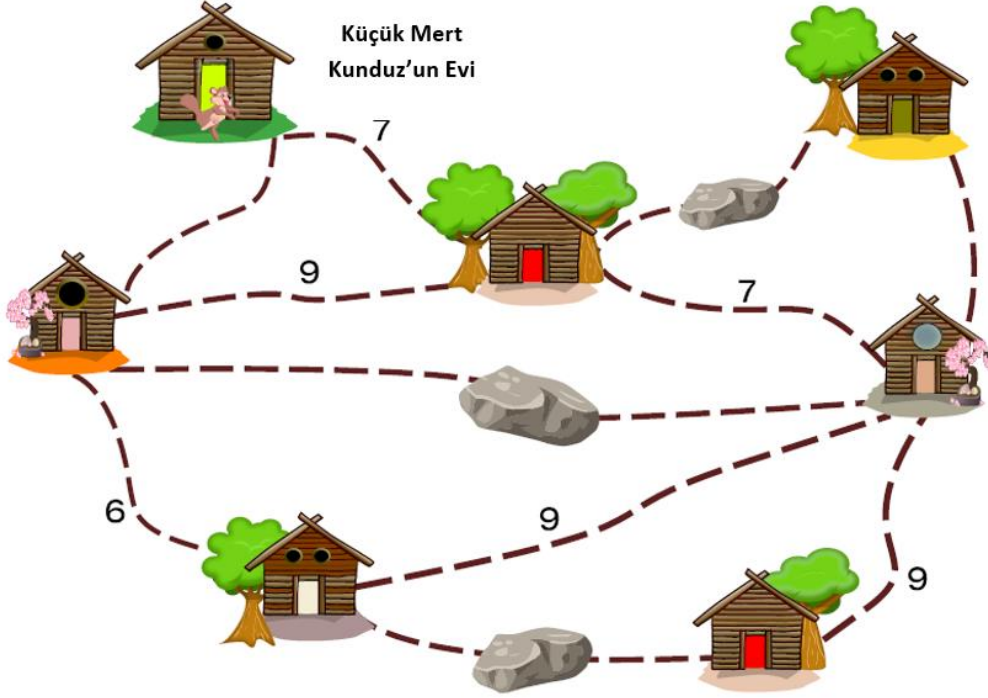
Kriptoloji, şifre metni, veri güvenliği

## İlgili Web Siteleri

<https://en.wikipedia.org/wiki/Cryptography>

## Ziyaretler

Küçük Mert Kunduz tatilde ve bütün akrabalarını ziyaret etmek istiyor. Yolların bazılarını kullanmak için ücret ödemesi gerekir (aşağıdaki şekilde görülmektedir). Bir yolu birden fazla kullanıyorsa, tekrar ücret ödemesi gerekmiyor. Yolların bazıları kayalar tarafından engellendiği için kullanılamazlar.



### Soru

Küçük Mert Kunduz'un tüm akrabalarını ziyaret edebilmesi için en az ne kadar paraya ihtiyacı vardır?

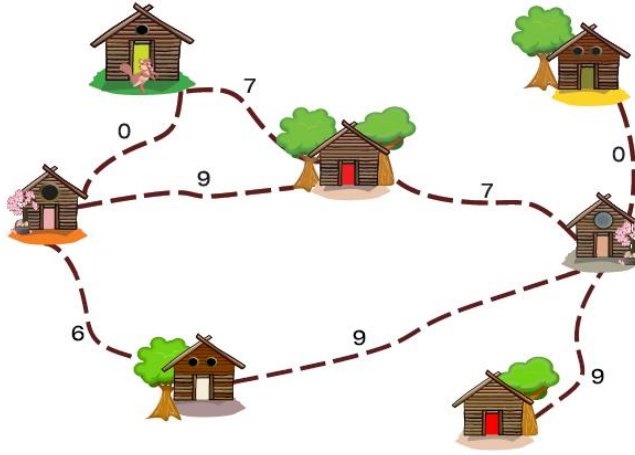
- A) 25
- B) 27
- C) 29
- D) 32

## Doğru Yanıt

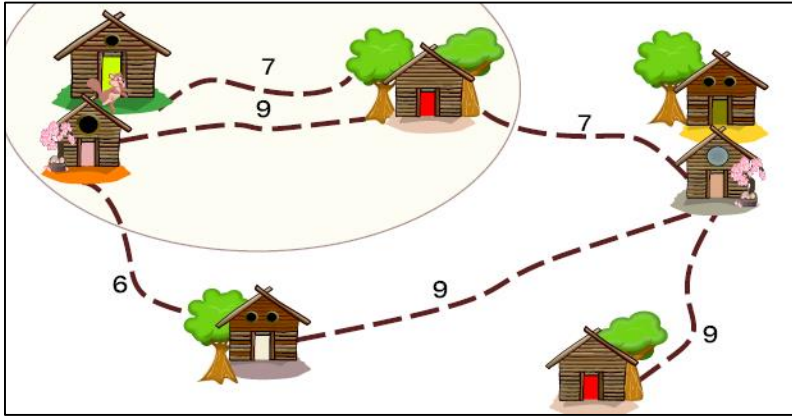
C

### Sorunun Çözümü

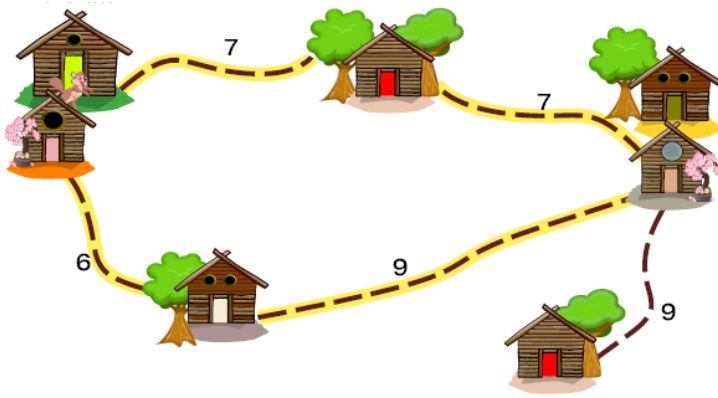
Ücreti olmayan yolların maliyetinin 0, kayalıklı yolların mevcut olmadığını düşünebiliriz. Böylece, harita şöyle görünecek:



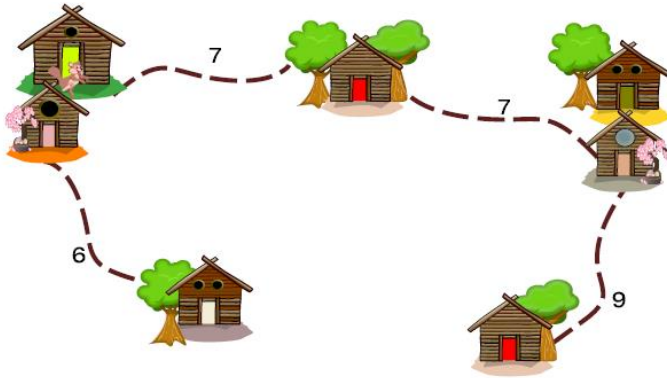
Yolların iki ucunda yer alan ve 0 ücretli yerler, aralarında rahatça dolaşabileceğimiz tek yer sayılabilir. Onları bir araya getirdikten sonra harita şöyle görünecektir:



Şimdi daire içine alınmış yerlerin aralarında iki yol olduğunu görüyoruz, bu nedenle daha yüksek ücretli olanı kaldırabiliriz.



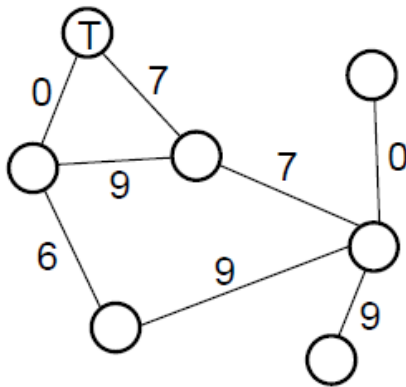
Çözmemiz gereken tek şey haritada vurgulanan devre. Devrede herhangi bir tek yolu kaldırabilir ve gerekirse diğer tüm evlere yine de başka yollarla ulaşabiliriz. Böylece devrede en yüksek ücreti alan yolu kaldırıyoruz.



Şimdi haritada kalan yolların ücretlerini toplamamız gerekiyor, yani  $6 + 7 + 7 + 9 = 29$

### Sorudaki Enformatik Kavramı

Genellikle bilgisayarlar, yerler arasında seyahat etmenin en iyi yolunu bulmak için kullanılır. 'En iyi', 'en hızlı' veya 'en kısa' veya 'en ucuz' anlamına gelebilir - bu görevde olduğu gibi - ve "yer", "şehir" veya "ev" veya "ağdaki bilgisayar" anlamına gelebilir. Bu tür sorulara en kısa yol problemleri veya daha genel olarak optimizasyon problemleri diyoruz.



Bilgisayar uzmanları sorunumuza Minimum Yayılma Ağacı (MST) problemi diyorlar. Minimum Yayılma Ağacı, tüm köşeleri mümkün olan en düşük maliyetle birleştiren kenarların bir alt kümesidir. MST'yi bulmanın klasik bir yolu Kruskal Algoritması.

## **Anahtar Kelimeler**

grafik, minimum maliyet yolları

## **İlgili Web Siteleri**

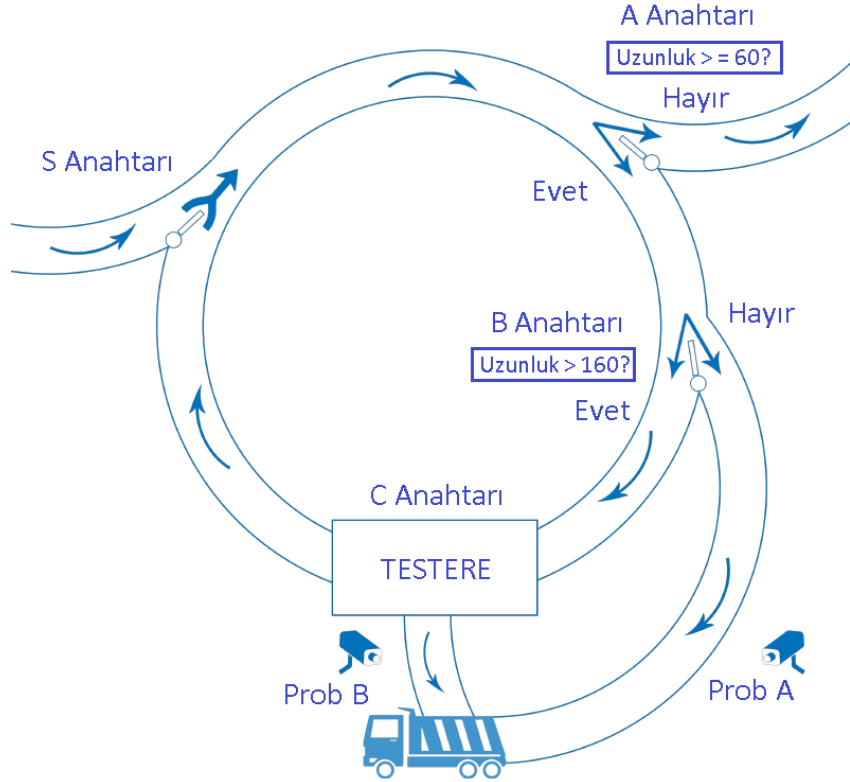
[https://en.wikipedia.org/wiki/Minimum\\_spanning\\_tree](https://en.wikipedia.org/wiki/Minimum_spanning_tree)

[https://en.wikipedia.org/wiki/Kruskal%27s\\_algorithm](https://en.wikipedia.org/wiki/Kruskal%27s_algorithm)

## Kereste Fabrikası

Bir kütük evi inşa etmek için doğru uzunlukta kütüklere ihtiyacınız vardır. Kütükler çeşitli uzunluklarda gelir ve aşağıdaki fabrika her bir tomruk kütesini 60 ila 160 cm uzunluğunda bir kamyonu koyar. Kayıtlar fabrikanın soluna gelir. Fabrikanın farklı kısımları şekilde görülmektedir:

- S anahtarı otomatik olarak kütükleri iki akıştan birine öncelikli olarak yönlendirir.
- A anahtarı, uzunluğu  $\geq 60$  cm olan kütüklerin geçmesini ve kısa kütüklerin gönderilmesini sağlar.
- B anahtarı, uzunluğu  $> 160$  cm olan kütüklerin geçmesine izin verir ve kalanları kamyonu gönderir.
- C anahtarı, kütüğü ikiye bölmek için testereyi çalıştırır. İlki 160 cm uzunluğunda olacak ve derhal kamyonu gönderecek. Kalan kütük diğer işlem için gönderilir.
- Prob A ve Prob B, kamyonu gönderilen kütük sayısını saymak için kullanılır.



### Soru

Fabrikaya üç farklı büyüklükte kütük (60 cm, 140 cm ve 360 cm) gönderilir. Tüm kütükler tamamen işlendiğinde, problemlerin saydığı değer nelerdir?

- A) Prob A: 1 kütük, Prob B: 3 kütük
- B) Prob A: 3 kütük, Prob B: 1 kütük
- C) Prob A: 2 kütük, Prob B: 2 kütük
- D) Prob A: 0 kütük, Prob B: 4 kütük

## Doğru Yanıt

C

### Sorunun Çözümü

Hiçbir kütük fiziksel olarak bir araya getirilmediğinden kütüklerin sırası önemli değildir.

- 60 cm'lik kütük, A anahtarını geçer. B anahtarı, kamyona gönderir. Prob A bir kütüğü sayar.
- 140 cm'lik kütük, A anahtarını geçer. B anahtarı, kamyona gönderir. Prob A bir kütüğü sayar.
- 360 cm kütük, A ve B Anahtarını geçer. C anahtarı, 160 cm ve 200 cm uzunluğunda iki kütüğe keser. 160 cm'lik kütük kamyona gönderilir ve Prob B bir kütüğü sayar.
- 200 cm'lik kütük, S, A ve B anahtarlarından tekrar geçer. Anahtar C, uzunluğu 160 cm ve 40 cm olmak üzere iki kütüğe keser. 160 cm'lik kütük araca gönderilir ve Prob B bir kütüğü sayar. 40 cm'lik kütük S anahtarından tekrar geçer, ancak A anahtarında çok kısa olduğu için gönderilir.

Sonuç olarak, Prob A ve B, her biri 2 kütük saymış olacak.

### Sorudaki Enformatik Kavramı

Akışın içinden geçen kütükler bir tamsayı dizisi olarak soyutlanabilir. Bu görev daha sonra, bir tam sayı akışının (kütüklerin uzunlukları) birkaç reaktif operatörle (birleştirme, filtreleme, dönüştürme) işlendiği ve bazı önlemlerin taramayla yapıldığı, bir Reaktif Programlamanın doğrudan uygulaması haline gelir.

S anahtarı, iki akışı birleştiren bir birleştirme işlemi olarak kabul edilebilir. A ve B anahtarları, her biri bir eşik değerini temel alarak bir akışı ikiye bölen filtre işlemleri olarak kabul edilebilir. Anahtar C, tam sayıları tam sayı çiftlerine dönüştüren bir dönüştürme işlemi olarak kabul edilebilir. Problar, geçen kütükleri sayan tarama operatörü olarak düşünebiliriz.

Örneğin, soruna bir çözüm RxJ'ler ile tanımlanabilir (<https://stackblitz.com/edit/rxjs-gx29ul> veya [2019-BE-06 Lumber Mill.js](https://2019-BE-06.Lumber.Mill.js)).

### Anahtar Kelimeler

Reaktif programlama, algoritma, akış

### İlgili Web Siteleri

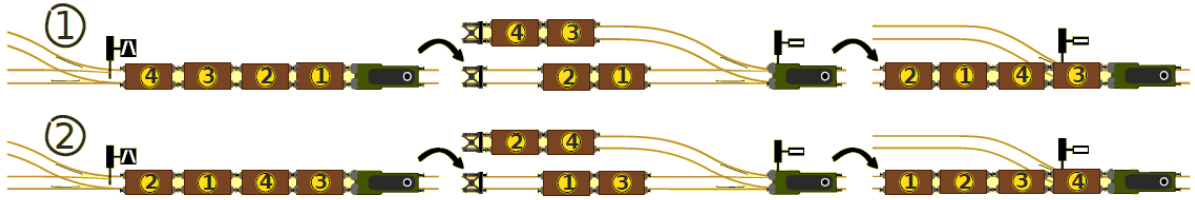
[https://en.wikipedia.org/wiki/Reactive\\_programming](https://en.wikipedia.org/wiki/Reactive_programming)

## Sınıflandırma Alanı

Bir tren, ana hat boyunca bireysel vagonları bırakmak zorundadır. Şu anda bir sınıflandırma bahçesine park edilmiştir ve ayrılmadan önce demiryolu araçları, en soldaki araba 1 1 numara olacak şekilde sıralanmalıdır.

Sınıflandırma sahası, tüm demiryolu araçlarının sağdan sola itildiği ve demiryolu araçlarının iki taraftan birine sarıldığı bir kambur sunmaktadır. Tüm demiryolu araçları kamburun üzerine itildikten sonra, lokomotif tarafından herhangi bir sırayla toplanırlar. Bütün bu süreç "bir işlem" olarak kabul edilir.

Örneğin, 4 demiryolu vagonunuz varsa, bunları sıralamak için onları iki kez (adım ① + ②) kamburun üzerine itmeniz yeterlidir:



Bunları kamburun üzerinde sadece bir işlemde sıralamak mümkün değildir.

### Soru

Demiryolu vagonları "2-8-3-1-5-7-6-4" sırasına sahipse, vagonları "1-2-3-4-5-6-7-8" şeklinde yerleştirmek için vagonların üst kısmına itilmesi gereken minimum sayı kaçtır?



- A) 1
- B) 2
- C) 3
- D) 4

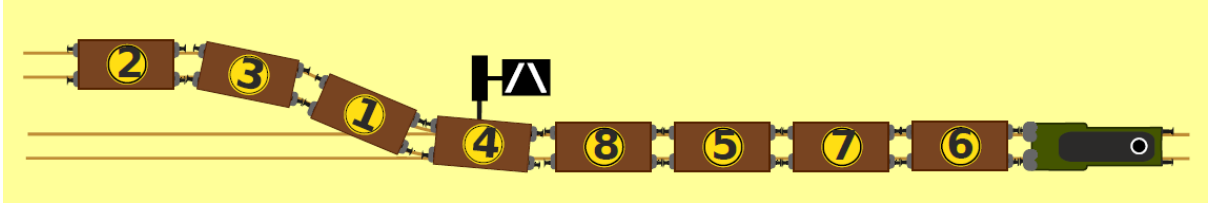


## Doğru Yanıt

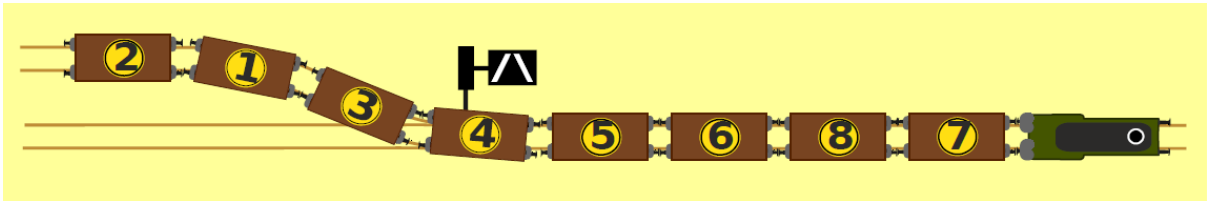
C

## Sorunun Çözümü

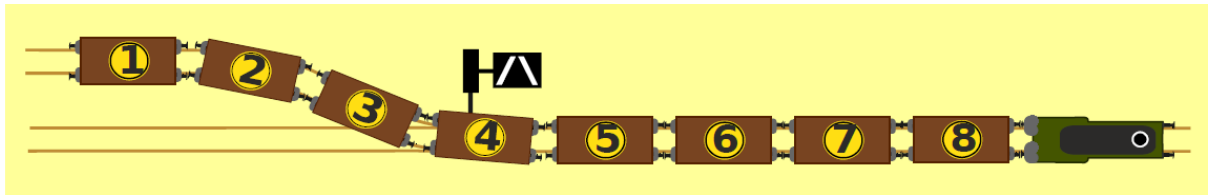
Demiryolu vagonlarını sıralamanın birçok yolu vardır; ancak en iyi yöntemlerden biri ilk önce 1 - 4 rakamı olan tüm vagonları üst tarafa, 5 - 8 rakamları olan vagonları alt tarafa itmektir. Henüz orada sıralanmayacaklar; ancak önce alt tarafı, sonra üst tarafı çekerseniz, aşağıdaki sıralamayı yaparsınız:



Şimdi 2, 1, 5 ve 6 no'lu vagonları üst tarafa ve 3, 4, 8 ve 7 no'lu vagonlara alt tarafa doğru iterek ve bunları dışarı doğru çekerek 5 ile 8 no'lu vagonların ilk önce dışarı çekilmesine rağmen her zaman aşağı doğru çekerek aşağıdaki sıralamayı yaparsınız:



Sonunda, tek sayılı demiryolu vagonlarını üst tarafa, çift numaralı tren vagonlarını alt tarafa iterek ve onları dışarı doğru çekerek, tek ve çift bir arabanın dönüşümlü olarak dışarı çekilmesini sağlayın, sonunda:



Demiryolu vagonları daha hızlı bir şekilde sıralanamamaktadır, çünkü vagon 4'ün 8'in önünde olması gerekir (bir kez kamburun üstünde), vagon 6'nın 8'in önünde ama 4'ten sonra (iki kez kamburun üzerinde olması gerekir, çünkü bu kamburun bir itişinde yapılamaz) ve vagon 7'nin 6 ile 8 arasında olması gerekir (kamburun üç katı, çünkü ilk kez 8'in önüne getirilebilir, ancak 4 7'nin önüne getirilemez; ikinci kez 6, 7 önünde hareket ederse, 4 7 önünde hareket edemez).

## Sorudaki Enformatik Kavramı

Tüm dünyadaki demiryolları bu sorunu günlük olarak yaşamaktadır, çünkü vagonların taşınması ve düzenlenmesi sıkıcı ve emek yoğun bir iştir: bir insan vagonları her seferinde birleştirmeli ve ayırmalıdır. Bu işlem zaman alır ve ana hattı bloke eder, özellikle lokomotif bir vagonu bir tarafa teslim ederken, bazı arabaların geride bırakılması ve ana hatta sabitlenmesi gerekiyorsa. Bu nedenle, daha büyük demiryolları birçok tarafı olan büyük sınıflandırma alanları geliştirmiştir. Bu görev sadece iki taraf sunuyor. Daha büyük trenler için bu tür sorunları çözmek zordur, ancak daha küçük demiryollarında, özellikle de daha büyük bir demiryolu ağına doğrudan bağlantısı olmayan dar ölçekli bir demiryolu ise, ortak bir durumdur.

Bilgisayar bilimi, bu vagonları ayırmada verimli olmak için çok yardımcı olmaktadır. Bu durumda, problemin bölümlerini tekrar tekrar çözmek çok basit hale getirir: bilgisayar bilimlerinde “böl ve yönet” adı verilen bir yöntem. Bu durumda, ilk operasyonda 8 arabayı sıralama problemini 4 arabayı sıralama problemine ayırıyoruz. Vagonları sıralamak için kullanılan yanlar, bilgisayar bilimlerinde yaygın olarak kullanılan soyut veri tipi yığını gibi çalışır. İzin verilen tek işlem “itme” ve “çekme” dir.

## Anahtar Kelimeler

böl ve yönet, yığın, sıralama

## İlgili Web Siteleri

[https://en.wikipedia.org/wiki/Divide-and-conquer\\_algorithm](https://en.wikipedia.org/wiki/Divide-and-conquer_algorithm)

[https://en.wikipedia.org/wiki/Classification\\_yard](https://en.wikipedia.org/wiki/Classification_yard)

[https://en.wikipedia.org/wiki/Stack\\_\(abstract\\_data\\_type\)](https://en.wikipedia.org/wiki/Stack_(abstract_data_type))

## Söğütler ve Kavaklar

Bilgin Kunduz sıcak yaz günlerinde gölge yapması için bahçeye ağaç dikmek istiyor. Söğüt ve kavak ağaçları arasında seçim yapmak için hangisinin daha hızlı büyüdüğünü bilmek istiyor. Daha fazla bilgi bulmak için tabletinde Arama uygulamasını kullanıyor. Uygulama, daha karmaşık arama sorguları için iki özel sembol kullanmaktadır:

... # ...	# İşaretinin her iki tarafındaki sorguları birleştirir. Yalnızca her iki sorguyu da içeren sayfaları gösterir.
[ ... ]	Sorguyu köşeli ayraçlar arasında ters çevirir. Yalnızca sorguyu karşılamayan sayfaları gösterir.

Bazı örnekler:

Sorgu	Anlamı
söğüt	Söğütler hakkında sayfalar
[ söğüt ]	Söğütler konusunu içermeyen sayfalar
söğüt # kavak	Hem söğüt hem de kavak ile ilgili bilgi içeren sayfalar
[[ kavak ]]	Kavak ile ilgili bilgi içeren sayfalar
[[ söğüt # kavak ]]	Ne söğüt ne de kavak ile ilgili bilgi içermeyen sayfalar
söğüt # [ kavak ]	Söğütle ilgili bilgi içeren ama kavak ile ilgili bilgi içermeyen sayfalar

### Soru

Aşağıdaki sorgulardan hangisi **sadece söğütlerle ilgili veya kavaklar** hakkında ya da **hem söğütler hem de kavaklar** hakkında bilgi içeren sayfaların tümünü bulmak için kullanılır?

- A) [söğüt] # [kavak]
- B) [[söğüt] # kavak] # [söğüt # [kavak]]
- C) [[söğüt] # [kavak]]
- D) [[söğüt] # [kavak]] # [söğüt # kavak]

## Doğru Yanıt

C

## Sorunun Çözümü

[[söğüt] # [kavak]]

Bunun neden doğru olduğunu görmek için önce biraz daha basit olan A yanıtını inceleyelim: '[söğüt] # [kavak]'. Bu sorgu hem '[söğüt]' hem de '[kavak]'ı karşılayan tüm sayfaları, diğer bir deyişle, söğütler hakkında bilgi içermeyen ve aynı zamanda kavaklar hakkında bilgi içermeyen tüm sayfaları bulmak ister – kısaca ne söğütten ne de kavak bahseden sayfalar.

Bu kesinlikle doğru değil, aslında bu Bilgin Kunduz'un aradığı şeyin tam tersi. Ama 'zıt' '[...]' anlamına geliyor. Yani, cevap A'yı köşeli parantezlerin arasına koymak tam olarak aradığımız şey - ve bu da yanıt C. '[söğüt # kavak]' sorgusu, hem söğütler hem de kavaklar hakkında bilgi içeren sayfaların bildirilmemesi gerektiğini belirtir. Başka bir deyişle: D yanıtı, Bilgin Kunduz'un aradığını, hem söğütler hem de kavaklar hakkında bilgi içeren sayfalar dışında gösterir. Bu nedenle, D sorgusu söğütler veya kavaklar hakkında bilgileri olan, ancak her ikisinde de olmayan sayfalara yönlendirecektir. B sorgusunun ne aradığını anlamak için onu iki bölüme ayırırız. '[söğüt] # kavak]' ilk bölümü kavaklarla ilgili olan ve söğütlerle ilgili olmayan sayfaları istemediğimizi, bir başka deyişle, sayfa kavaklarla ilgiliyse, aynı zamanda söğütlerle ilgili olması gerektiğini belirtir. Benzer şekilde, ikinci bölüm '[söğüt # [kavak]]', sayfa söğütler hakkında bilgi içeriyorsa, ayrıca kavaklar hakkında da bilgi içermesi gerektiğini söyler. Her iki gereksinimin birleştirilmesi (= sorgu B), hem söğütler hem de kavaklar ya da hiçbiri hakkında olmamak zorunda olan sayfaları verir.

## Sorudaki Enformatik Kavramı

Sanal Arama uygulaması gibi, modern arama motorlarımız da, DEĞİL, VE veya VEYA kullanarak sorguların birkaç şekilde birleştirilmesine izin verir. Bilgin Kunduz'un gerçekten istediği şey söğütleri VEYA kavakları aramak, fakat ne yazık ki Arama uygulamasında 'VEYA' sembolü yok. Bu görev, 'VEYA'nın, özellikle' DEĞİL'lerin ve 'VE'lerin bir kombinasyonu kullanılarak ifade edilebileceğini gösterir.

a VEYA b = DEĞİL (DEĞİL (a) VE DEĞİL (b))

Matematik alanında bu De Morgan'ın yasalarından biri olarak bilinir.

Programlama sırasında birçok mantıksal ifade kullanıyorsunuz. De Morgan yasaları gibi bazı temel yasaları bilmek, karmaşık programları yazmayı ve anlamayı kolaylaştırır. 'VEYA' (OR) kelimesine dikkat etmeliyiz çünkü farklı anlamları olabilir. Yanlış yanıtlardan biri -

seenek D - OR (ya da XOR) olarak adlandırdığımız şeyi döndürür: ya söğütler ya da kavaklar hakkındaki web sitelerini, ancak ikisini birden değil. Bu günlük hayatta sıklıkla kullandığımız VEYA'dır. Birisi size patates veya pizza yemek isteyip istemediğınızı sorarsa, genellikle ikisini de yemek istediğınızı söylemezsiniz.

Seenek C söğütler veya kavaklar veya her ikisi hakkında sayfaları verir. Son bir not: Arama uygulamasının AND için '#' sembolünü kullanmasını ve belki de 'VE' kelimesini veya '&' sembolünü ve NOT için köşeli parantezleri kullanmasını garip bulabilirsiniz. Bilgisayar için bu fark yaratmaz, sadece programlanması gerekenleri yapar.

### **Anahtar Kelimeler**

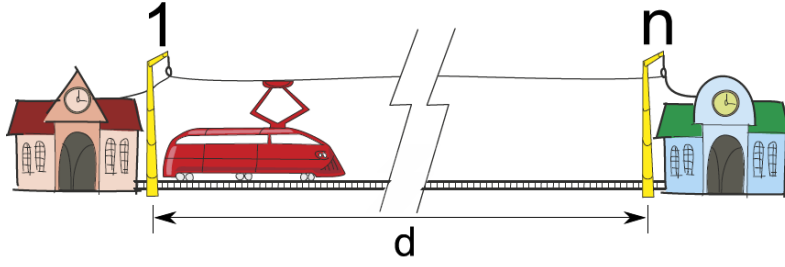
De Morgan Yasası

### **İlgili Web Siteleri**

[https://tr.wikipedia.org/wiki/De\\_Morgan\\_yasas%C4%B1](https://tr.wikipedia.org/wiki/De_Morgan_yasas%C4%B1)

## Tren Yolunda Elektriklenme

İki istasyon arasındaki demiryoluna elektrik hattı döşenmesi gerekiyor. Elektrik direkleri, ilki birinci istasyonda ve sonuncusu 2. istasyonda olacak biçimde düzenli aralıklarla yerleştirilmelidir. Direk yerleştirme robotu sağdaki programa göre çalışır. İstasyonlar arasındaki mesafe “d” metredir ve robot “n” adet direk yerleştirmelidir.



tanımla - direk yerleştir
depoya sür
X tane direk yükle
İlk istasyona git
bir direk yerleştir
tekrarla Y
Z metre ileri git
bir direk yerleştir
↻

### Soru

Robotun doğru çalışması için programa hangi parametrelerin (X, Y ve Z) eklenmesi gerekir?

- A) n, n-1, d/(n-1)
- B) n, n-1, d/(n+1)
- C) n-d, n+1, d/(n-1)
- D) n/2, n-1, d/(n+1)

## Dođru Yanıt

A

### Sorunun Çözümü

$n, n-1, d/(n-1)$

Robotun  $n$  adet direk yerleřtirmesi için  $n$  direk yüklemelidir. Bir direk yerleřtirildikten sonra,  $n-1$  direk kalır. Böylece robot  $n-1$  kez taşıma ve yerleřtirme hareketini tekrarlamalıdır. O zaman robotun bir sonraki diređi yerleřtirmek için ne kadar hareket etmesi gerektiđini bilmek için mesafeyi  $(n-1)$  ile bölmeliyiz.

### Sorudaki Enformatik Kavramı

Bu görevdeki kod parçası bir işlem grubudur. Bu işlem grubunda  $d$  ve  $n$  deđişkenlerini kullanıyoruz. Bir deđişken, bir deđer için bir isim olarak kabul edilir, ör. bazı numaralar. Bu deđişkenler ( $d$  ve  $n$ ) daha önce ana programda tanımlanmış ve bu işlem grubunda kullanılmıştır. Bunlara global deđişkenler denir, çünkü aynı programın farklı işlem gruplarından erişilebilirler. Görev de döngülerle ilgilidir. Döngü, birkaç kez yürütülen yönergeler dizisidir. Bu görevde, döngünün kaç kez yürütülmesi gerektiđini bilmek çok önemlidir. Bizim görevimizde, demiryoluna elektrik hattı döşenmesini tamamlamak için döngü  $n-1$  kez tekrarlanmalıdır. İşlem grupları, yani sıralı program yönergeleri, belirli bir görevi yerine getiren güçlü programlama araçlarıdır. Deđerler işlemlere deđişkenlerden geçerek iletilir. Aynı işlem grupları deđişkenlere farklı deđerler geçirerek farklı işlemler üzerinde çalışabilir.

### Anahtar Kelimeler

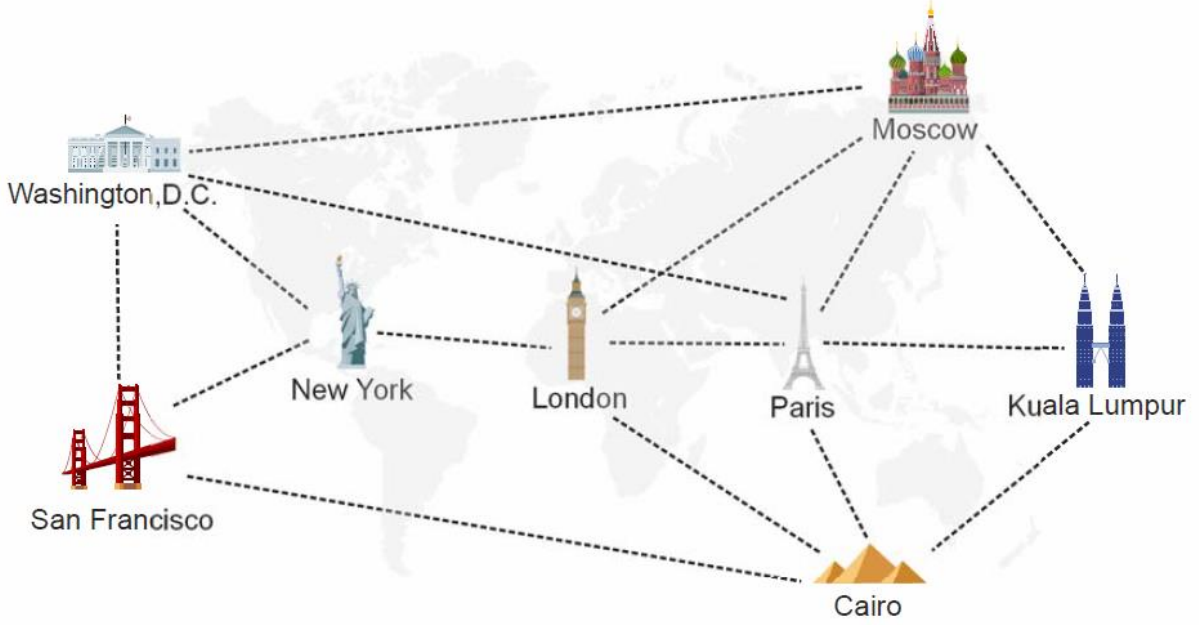
Programlama, parametrelerle işlemler, deđişkenler

### İlgili Web Siteleri

<https://tr.wikipedia.org/wiki/De%C4%9Fi%C5%9Fken>

## Yeşil Uçuş Rotaları

Kunduz Havayolu, resimde gösterildiği gibi, dünyadaki birçok büyük şehri birbirine bağlayan çok sayıda uçuş rotasına sahiptir.



Karbon ( $CO_2$ ) emisyonları küresel ısınmanın önemli bir nedenidir. Havayolu, karbon ( $CO_2$ ) emisyonlarını azaltmak için, müşterilerin herhangi bir şehre uçma olanağını yok etmeden bazı uçuş rotalarını iptal etmek istiyor.

Örneğin, San Francisco ve Washington, D.C. arasındaki uçuş rotası iptal edilirse müşteriler San Francisco'dan New York'a, daha sonra New York'tan Washington D.C.'ye uçabilirler.

### Soru

Yukarıda gösterilen uçuş rotaları için havayolunun iptal edebileceği maksimum rota sayısı nedir?

- A. 6
- B. 7
- C. 8
- D. 9

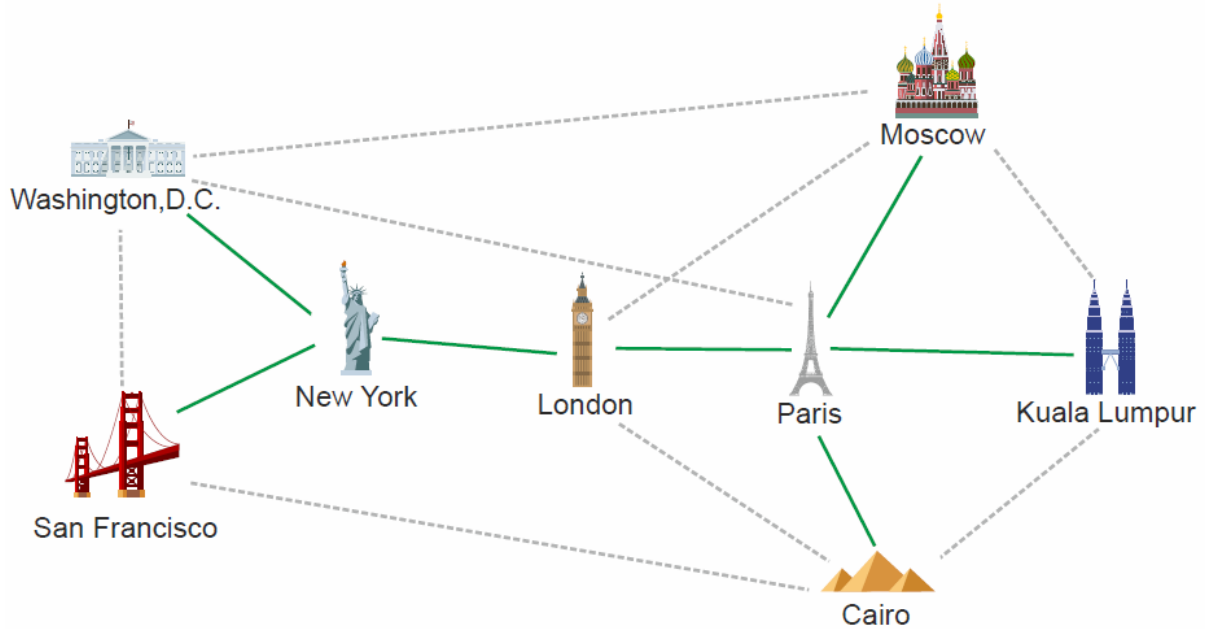


## Doğru Yanıt

C

### Sorunun Çözümü

Aşağıdaki örnek, 8 rota iptal edilse bile müşterilerin istedikleri şehre uçabileceğini göstermektedir.



8 rota bıraktığımızda  $15 - 8 = 7$  rota kaldığına dikkat edin. Neden daha fazla rota bırakılamıyor? Sadece 2 şehir olsaydı, 1 rotaya ihtiyacımız olacağı açıktı. 3 şehir olsaydı 2 rotaya ihtiyacımız olurdu. Bu model devam eder ve 8 şehir olduğunda 7 rotaya ihtiyacımız olduğunu görürüz.

Modelin doğru olduğundan emin olmak için, 9 rota (veya daha fazla) iptal ederseniz, müşterinin her zaman bir şehre uçamayacağına kendimizi ikna etmemiz gerekir. 9 rota (veya daha fazla) iptal edildiğinde, sadece 6 rota (veya daha az) olacaktır. Yalnızca 6 yol varsa ne yanlış gider? Yalnızca 6 güzergah varsa, 8 şehir olduğundan, herhangi bir güzergahta olmayan bir şehir veya tam olarak bir güzergahta bir şehir vardır. Tam olarak bir rotada bir şehir varsa, o zaman bu şehri ve rotayı kaldırabilir ve 5 rota ve 7 şehre bırakılabilir. Bu mantıkla devam edersek, her durumda, rotada olmayan bir şehir ile veya arasında rota olmayan 2 şehir ile karşılaşırız. Bu nedenle, yalnızca 6 güzergah varsa, müşteri her zaman her istediği şehre uçamayabilir.

### Sorudaki Enformatik Kavramı

Havayolu yolları bir grafik olarak gösterilebilir. Şehirler grafikteki düğümlerdir ve uçuş yolları kenardır. Tüm şehirlerin birbirine bağlı olmasını sağlarken maksimum rota sayısını bırakma problemi, bilgisayar bilimlerinde Minimum Yayılma Ağacı (MST) probleminin bir uygulamasıdır. Minimum Yayılma Ağacı sorunu, grafiğin tüm düğümlerini birbirine bağlayan bir ağacın çıkarılmasından ibarettir. Bilgisayar uzmanları telekomünikasyon ağları, ulaşım ağları veya su kaynağı ağlarının döşenmesindeki sorunları çözmek için Minimum Yayılma Ağacı algoritmalarını kullanır.

## **Anahtar Kelimeler**






















grafik, yayılan ağaç

## **İlgili Web Siteleri**

[https://en.wikipedia.org/wiki/Minimum\\_spanning\\_tree](https://en.wikipedia.org/wiki/Minimum_spanning_tree)

## Arı Kovanı

Bir arıcıda, arı kovanı vardır. Arıcı, kovanı öyle bir noktaya yerleştirmek istiyor ki, kovandan her çiçeğe kadar olan mesafelerin toplamı minimum olsun. Çiçekli alan, aşağıdaki tabloda, 1'den 9'a kadar olan satırlar ve A'dan I'ya kadar olan sütunlarla gösterilmiştir.

	A	B	C	D	E	F	G	H	I
1									
2									
3									
4									
5									
6									
7									
8									
9									



Arılar sadece bu alanda yatay ve dikey olarak uçarlar; bu nedenle iki hücre arasındaki mesafe yatay mesafenin ve dikey mesafenin toplamıdır. Örneğin, C4 ve D7 arasındaki mesafe 4'tür (dikey olarak 3 hücre artı yatay olarak 1 hücre).

### Soru

Arıcı, kovanı nereye koyarsa kovandan her çiçeğe kadar olan mesafe minimum olur? (olası

yerler haritada  olarak işaretlenmiştir)

- A) D5
- B) C7
- C) E5
- D) A9

## Dođru Yanıt

A

### Sorunun özümü

Medyan ieđinin sırasını dikey olarak ve medyan ieđinin sütünunu yatay olarak bulma sorunu azaltılabilir. Tarlada 17 iek olduđundan, medyan iek dikey ve yatay olarak 9. sıradadır (bir tarafta 8, diđer tarafta 8 iek vardır). Dikey olarak sıralandıđında, ortanca ieđi A5'tir, bu nedenle kovanın 5. sıraya konması gerekir. Yatay olarak sıralanırsa, ortanca ieđi ya D3 ya da D9'dadır; ünkü her ikisi de aynı sütundadır. Bu nedenle kovanın D sütununa konması gerekir. Bu bize kovanın D5 olması için en uygun yeri verir.

Uzaklık dikey ve yatay mesafenin toplamı olarak kabul edildiđinden (Öklid, düz izgi, mesafe deđil), sorun, kovanı bir koordinat üzerinde hareket ettirmeden bađımsız olarak sırayı ve sütünu bulmakla bölünebilir diđer koordinattaki mesafeleri etkiler. Bu nedenle, en uygun sütünu bulmak için kullanılan aynı algoritma, en uygun sırayı bulmak için de geçerlidir ve en uygun konum, en uygun sütundan ve en uygun sıradan oluşacaktır.

### Sorudaki Enformatik Kavramı

Bu problemi özmenin amaçlanan yolu (yani tüm yolun mesafelerini saymamak) böl ve yönet algoritması tasarım paradigmasını kullanır. ünkü temel problemi iki alt soruna böler: optimal sırayı bulmak ve en uygun sütünu bulmak. İki alt sorunu özdükten sonra, sonuçlar birleştirilir.

### Anahtar Kelimeler

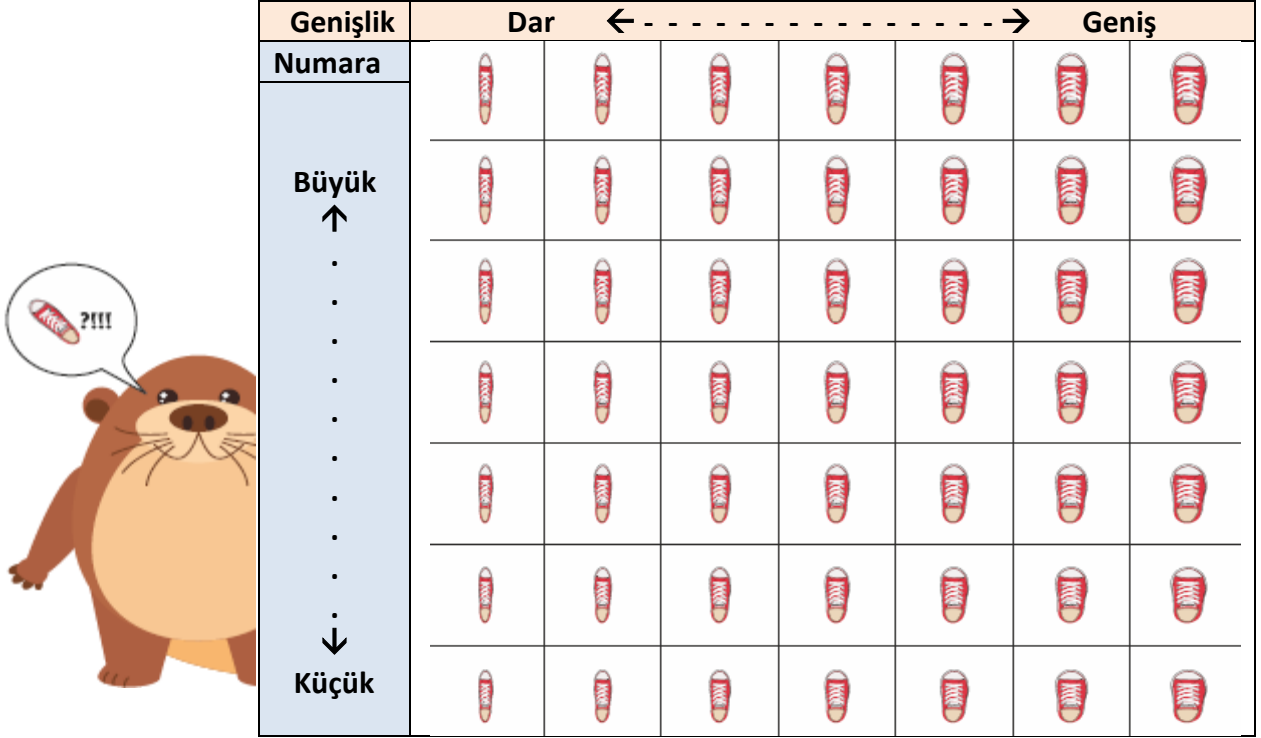
Medyan, L1 mesafesi, Manhattan mesafesi

### İlgili Web Siteleri

[https://en.wikipedia.org/wiki/Divide-and-conquer\\_algorithm](https://en.wikipedia.org/wiki/Divide-and-conquer_algorithm)

## Ayakkabı Boyutu

Kunduz Çınar, bir çift ayakkabı almak için ayakkabı dükkânına gitti. Resimde gösterildiği gibi ekranda birkaç ayakkabı gördü. Ayakkabılar, numara ve genişlik sırasına göre düzenlenmişti. Ayakkabılar, en küçük numara ve dar ayakkabılar sol altta, en büyük numara geniş ayakkabılar sağ üstte olacak biçimde düzenlenmişti. Tüm ayakkabıların farklı boy ve genişlikleri vardır. Unutkan bir kunduz olan Çınar, ayakkabı numarasını hatırlamıyordu. O yüzden doğru olanı bulana kadar ayakkabıları denemesi gerekecekti. Doğru bir uyum, doğru numara ve doğru genişlik demektir.



Genişlik	Dar ← - - - - - → Geniş						
Numara							
Büyük ↑							
·							
·							
·							
·							
·							
·							
Küçük ↓							

Ancak, Kunduz Çınar, 'n' denemede uygun ayakkabıyı bulmasını garanti eden bir yöntem kullanıyor.

### Soru

'n' değeri en küçük kaç olabilir?

- A) 2
- B) 3
- C) 4
- D) 5



Diyelim ki kunduzun denediđi ayakkabı çifti onun için daha küçük ve daha geniş. Bu yüzden bölge 1'de olacak daha büyük ve daha dar ayakkabıları denemesi gerekir. Şimdi bölge 1'in merkezindeki (kutu # 1 ile işaretlenmiş) ayakkabıyı deniyor.

- Eğer ayakkabı ona uyuyorsa, doğru ayakkabıyı buldu.
- Eğer ayakkabı hala daha küçükse ve onun için daha genişse, A pozisyonunda ayakkabı uygun olacaktır.
- Eğer ayakkabı daha küçük ancak doğru genişlikte ise, B konumundaki ayakkabı daha uygun olacaktır.

Gördüğümüz gibi, kunduz en uygun ayakkabıyı bulmak için en fazla 2 ayakkabıyı denemek zorunda kalacak. Başka bir pozisyonda başlarsa, daha fazla ayakkabımızı denemek zorunda kalır.

### **Sorudaki Enformatik Kavramı**

Yukarıdaki problemde, ayakkabı mağazasında sergilenen ayakkabılar, çoklu boyutta artan ebat ve genişlik sırasına göre düzenlenir. Böyle bir düzenlemeye sıralama denir. İkili arama algoritmaları, bir öğeyi çok hızlı bir şekilde bulmak için sıralanmış verilerde kullanılır. İkili arama, daha az denemede doğru cevabı almak için her seferinde arama alanını yarıya indirir.

Örneğin, ikili arama algoritmasını kullanarak, 1 ile 100 arasında bir sayı tutup arkadaşınızın bu sayıyı minimum kaç deneme sonra bulabileceğini tahmin edin.

### **Anahtar Kelimeler**

İkili arama algoritması, Mantıksal akıl yürütme

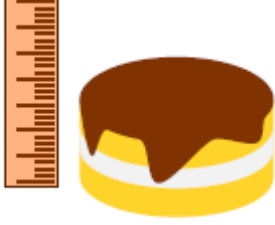
### **İlgili Web Siteleri**

[https://en.wikipedia.org/wiki/Binary\\_search\\_algorithm](https://en.wikipedia.org/wiki/Binary_search_algorithm)

[https://en.wikipedia.org/wiki/Logical\\_reasoning](https://en.wikipedia.org/wiki/Logical_reasoning)

## Kekler ve Komşular

Cuma sabahı üç komşu Anıl, Betül ve Çiler, her biri aynı pastaneden Cumartesi günü bir parti için bir pasta sipariş etti. Üçü de aynı türde, 3 cm boyunda olan bir kek sipariş etti.



Ancak, her biri siparişlerini değiştirmek için pastaneyi aradı. Pastacı her zaman yeni isteği yazar ve eskisini iptal eder. Kekler cumartesi sabah erken saate hazırlanacak.

- Anıl arar: “Pastamı sipariş ettiğimden 1 cm daha uzun yapın”.
- Daha sonra Anıl tekrar arar: “Şimdi, kekimi Betül ile aynı boyda yapın”.
- Betül arar: “Pastamı sipariş ettiğimden 2 cm daha uzun yapın”.
- Daha sonra “Betül tekrar arar: Şimdi pastamı sipariş ettiğimden 1 cm daha küçük yapın”.
- Çiler arar: “Pastamı Anıl’ın sipariş ettiği kekten 1 cm daha uzun yapın”.
- Daha sonra Çiler tekrar arar: “Pastamı sipariş ettiğimden 1 cm daha uzun yapın”.

Bu telefon görüşmelerinin hangi saatlerde yapıldığını bilmiyoruz. Ama her bir komşunun ikinci araması, ilk aramalarından sonra yapıldı.

### Soru

Cumartesi günü aşağıdaki ifadelerden hangisi doğrudur?

- A) Anıl ve Betül’ün pastası aynı yüksekliğe sahip.
- B) Betül’ün pastası Çiler’in pastasından en az 1 cm daha kısa.
- C) Çiler’in pastası Anıl’ın pastasından 2 cm daha uzun.
- D) Her üç pasta da en az 4 cm boyunda.



## Doğru Yanıt

B

### Sorunun Çözümü

Yanıt A doğru değil: eğer Betül, Anıl'ın çağrılarında sonra tüm çağrıları yaparsa, o zaman Anıl, 3 cm'lik ve Betül 4 cm'lik pasta alır. Bu aynı zamanda D yanıtının doğru olmadığını gösterir.

Yanıt C doğru değil: önce Çiler bütün görüşmelerini yapar, sonra Betül ve sonra Anıl görüşmelerini yaparsa, Çiler'in pastası 5 cm, Betül'ün pastası 4 cm ve Anıl'ın da 4 cm'lik olur. Betül'ün pastacıya verdiği talimatlara bakarsanız öğleden sonra bir noktada pastacının Betül'ün keki için 3 cm, 5 cm ve 4 cm yüksekliği yazdığını göreceksiniz. Sonuç olarak, Anıl'ın pastası için de pastacı yükseklik olarak 3 cm, 4 cm veya 5 cm yazmış olabilir. Bu, Cumartesi günü Çiler'in pastasının 5 cm, 6 cm veya 7 cm yüksekliğinde olacağı anlamına gelir. Betül'ün pastası her zaman 4 cm boyunda olacaktır. Yani B yanıtı doğrudur.

### Sorudaki Enformatik Kavramı

Bir bilgisayar, aynı uygulamada bile aynı anda birçok farklı şey yapabilir. Örneğin, bir sözcük için yazım denetimi siz sözcüğü yazarken çalışır. Hikâyenin gösterdiği gibi, farklı görevler aynı anda gerçekleştirildiğinde, sonuçların ne olacağını tahmin etmek her zaman kolay değildir. Bunun nedeni, her bir görevin tam olarak her bir bölümünün ne zaman yürütüleceğini tahmin etmenin genellikle zor olmasıdır.

Aynı anda yürütülen görevlerden oluşan programları yazmaya eşzamanlı programlama denir. Programcıların bu tür programlamaya dahil olan sorunların farkında olmaları ve belirli eylemlerin her zaman aynı sırada yürütüleceğinden emin olacak şekilde programlarını yazmaları gerekir. Örneğin, bir ağ yazıcısı için yazılım oluşturan programcılar (yaklaşık olarak) aynı anda yazıcıya gönderilen iki belgenin birbiri ardına yazdırılmasını sağlamalıdır.

Günümüzde, bilgisayarlar aynı anda birçok şeyi yapabilmek için özel olarak inşa edilmiştir. Eşzamanlı programlama ve yanlış yapmaktan kaçınmak için ilgili teknikler, son zamanlarda bilgisayar bilimleri öğretim programının bir parçası olarak büyük önem kazanmaktadır.

### Anahtar Kelimeler

Eşzamanlılık

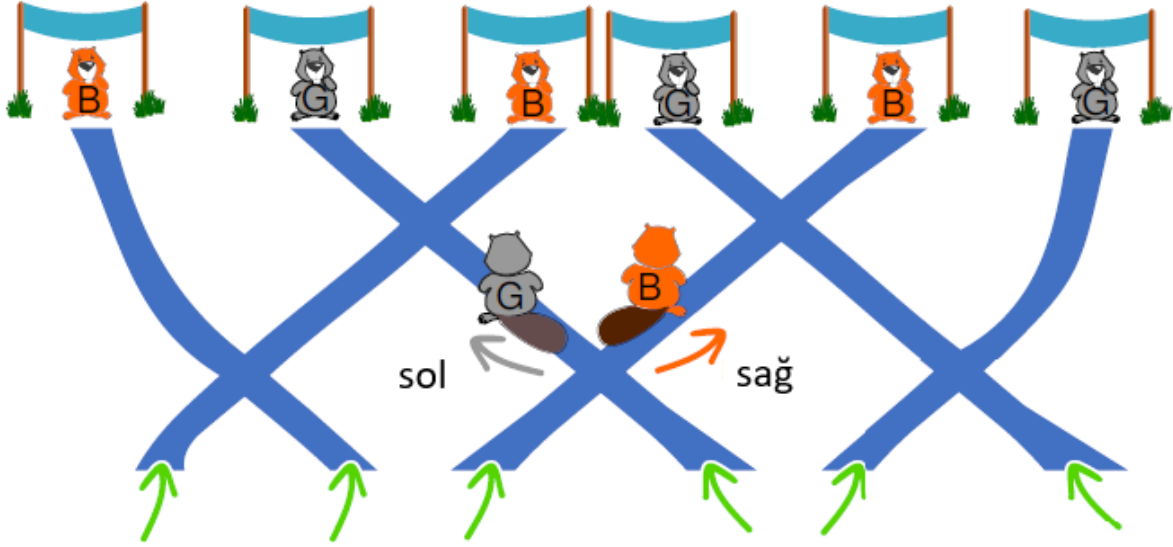
### İlgili Web Siteleri

[https://en.wikipedia.org/wiki/Concurrency\\_\(computer\\_science\)](https://en.wikipedia.org/wiki/Concurrency_(computer_science))

[https://tr.wikipedia.org/wiki/Paralel\\_zamanl%C4%B1l%C4%B1k](https://tr.wikipedia.org/wiki/Paralel_zamanl%C4%B1l%C4%B1k)

## Kunduz Ağı

Kunduzların girip çıktığı bir geçiş ağı vardır. Bu ağın, altı girişi ve altı çıkışı vardır. Her girişten yalnızca bir kunduz girebilir. G ve B olmak üzere iki tür kunduz vardır. İki kunduz bir kavşakta karşılaşır ve renkleri farklı ise, Kunduz G sola giderken, Kunduz B sağa gider. Altı kunduz aynı anda geçiş ağına girer.



### Soru

Kunduzlar ağdan B G B G B G olarak çıkıyorsa, hangi sırada girmişlerdir?

- A) B B G B G G ya da B B B G G G
- B) B G G B G B ya da B B B G G G
- C) B B G B G G ya da B B G B G G
- D) B G G B G B ya da B G B G G B

## Dođru Yanıt

A

## Sorunun özümü

En sol ıkıřta B olması için, en soldaki iki giriřin BB olması gerekir. Bařka kombinasyon en sol ıkıřta G üretir. En sađ ıkıřta G olması için, en sađ iki giriřin GG olması gerekir. Diđer herhangi bir kombinasyon en dođru ıkıřta B üretir. Giriřteki iki orta pozisyon için sadece iki kombinasyonla kalıyoruz: BG veya GB. Her ikisinin de dođru olduđunu kolayca kontrol edebiliriz.

## Sorudaki Enformatik Kavramı

Bu görev bazı kurallara göre bir ađda dolařan kunduzlarla ilgilidir. Bu, bilgisayar ađları üzerinden veri akıřına karřılık gelir. Buna IP yönlendirme denir. IP yönlendirme, farklı ađlar arasında veri paketlerini tařıma iřlemidir. Varsayılan olarak iki farklı IP ađı birbirleriyle iletişim kuramaz. Paketi aralarında deđiřtirebilen bir arabulucu cihaza ihtiyaları var. Bir yönlendirici bununla ilgilenir. Yönlendirici arayüzleri farklı ađlarla ilişkilidir. Bu bađlantı yönlendirme tablosunda tutulur. Yönlendiriciler anahtarlama kararını vermek için kullanırlar.

## Anahtar Kelimeler

İletişim, ađlar

## İlgili Web Siteleri

<https://www.computernetworkingnotes.com/ccna-studyguide/basic-routing-concepts-and-protocols-explained.html>

## Uçak Planlaması

Bir uçak havaalanına indiğinde, kazaları önlemek için uçaklara bir koridor atanır. Bu, uçakları birbirinden ayıran bir hava sahasıdır. Kunduz Adası havaalanında, iniş süreleri arasında 15 dakika yoksa iki uçak aynı koridorda olamaz.



Örneğin, Uçuş #1 sabah 6: 10'da, Uçuş# 2 sabah 6:25'te ve Uçuş #3 sabah 6: 26'da iniyorsa, Uçuş # 1 ve Uçuş #2 aynı koridora atanamaz. Uçuş #3 ve Uçuş #1 ile aynı koridora atanabilir, ancak Uçuş #2 ile aynı koridora atanamaz. Bugün havaalanında Hava Trafik Kontrolörü sizsiniz ve göreviniz tabloda gösterilen tarifeli uçuşlara koridorlar atamak.

Uçuş	İniş Zamanı
9V2400	7:00
9V1321	7:21
AI561	7:20
AI620	7:18
EK427	7:03
SG147	7:12

### Soru

Yukarıdaki tüm uçuşların kurallara uygun şekilde inmesini sağlamak için gereken minimum koridor sayısı nedir?

- A) 2
- B) 3
- C) 4
- D) 5

## Doğru Yanıt

C

### Sorunun Çözümü

Uçuşları ve uçuş saatlerini artan iniş sırasına göre sıralayalım.

İlk uçuş 9V2400 saat 7: 00'dedir, bu yüzden koridor 1'e atadık. Sonraki uçuşlardan 2. uçuş (EK427) ilk uçuştan 3 dakika sonra, 3. Uçuş (SG147) ilk uçuştan 12 dakika sonra ve 3. uçuş da 2. uçuştan 9 dakika sonra, açıkça üç ayrı koridor atanması gerekir. Şimdiye kadar 3 koridor lazım.

Aşağıdaki uçuşların her biri için, mümkünse zaten kullanılmış koridorlara atamaya çalışıyoruz. Saat 7: 18'deki AI620 numaralı dördüncü uçuş, ilk 9V2400 uçuşundan 18 dakika sonradır ve 9V2400 uçuşuyla aynı koridor kullanılabilir.

Benzer şekilde, 5. uçuş AI561 ile AI620 arasında 15 dakikadan az olduğu için koridor 1'e atanamaz, ancak EK427 ile arasında 15 dakikadan fazla olduğu için koridor 2'ye atanabilir. Saat 7: 21'deki 9V1321 numaralı uçuş, şu ana kadar 3 koridora atanan SG147 (7:12), AI620 (7:18) ve AI561 (7:20) uçuşları ile arası 15 dakikadan daha kısa bir süredir. Bu nedenle bu uçuş için başka bir koridor atamalıyız. Böylece koridorlara şu uçuş görevlerini veririz:

Koridor 1: 9V2400 (7:00), AI620 (7:18)

Koridor 2: EK427 (7:03), AI561 (7:20)

Koridor 3: SG147 (7:12)

Koridor 4: 9V1321 (7:21)

4'ten fazla koridor atayabilirsek de, 4'ten daha azının kazalara neden olabileceği açıktır.

### Sorudaki Enformatik Kavramı

Yukarıdaki problemde çatışmalarınız var (birbirlerinin 15 dakika içinde iniyorlarsa iki uçak aynı koridora sahip olamazlar) ve kaynaklar (koridorlar) ve çatışmaları çözecek biçimde kaynakları atamanız gerekir. Bu tür birçok problem var.

Probleminizde çok fazla nesne varsa (uçak, misafir,...), çabucak çözmek istersiniz. Bilişimde de bu problemleri çözmek için özel yaklaşımlar geliştirilmiştir. Bunlardan biri, onu temsil etmek için bir grafik kullanmaktır. Bir grafikte, düğümleri birbirine bağlayan düğümler ve kenarlar vardır. Bu durumda, düğümler nesnelere (uçuşları) ve kenarları çatışmaları temsil eder: iki nesne birbiriyle çatışıyorsa, düğümleri bir kenar ile bağlanır, aksi halde olmazlar. Görevin sorunu, bir grafik renklendirme sorunu olarak düşünülebilir: grafikteki her düğümü, bir kenar tarafından birbirine bağlanmış aynı renkli düğüm olmayacak şekilde renklendirmek için gereken minimum renk sayısı nedir?

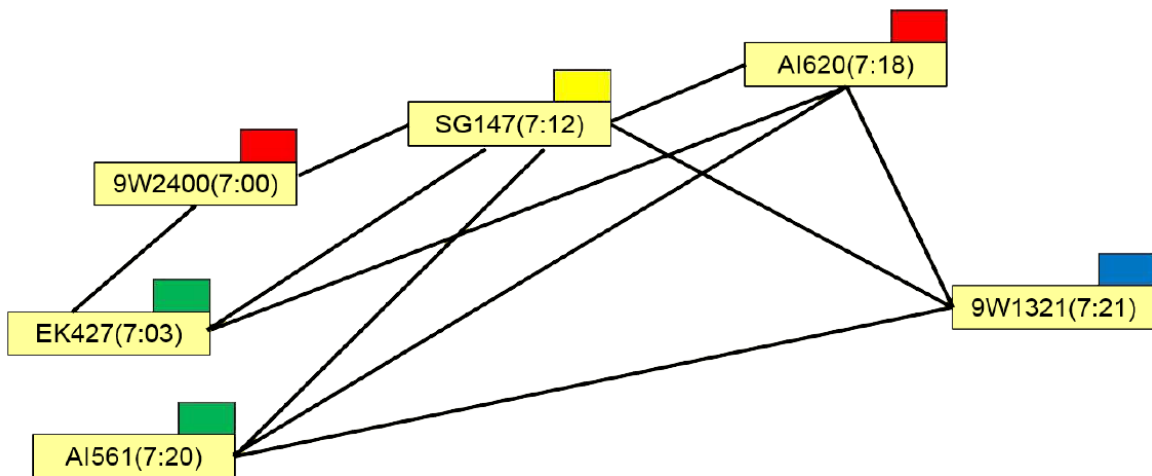
Uçak çizelgeleme problemine uygulandığında, her renk bir koridoru temsil eder. Grafik renklendirme, uçakların uçuş rotalarına programlanması, radyo istasyonlarına bant genişliği

Uçuş	İniş Zamanı
9V2400	7:00
EK427	7:03
SG147	7:12
AI620	7:18
AI561	7:20
9V1321	7:21

tahsisi, örüntü eşleştirme, spor planlaması, oturma planlarının tasarlanması, sınav çizelgeleme, kabinlerin programlanması ve hatta SudoKu bulmacalarının çözülmesi gibi uygulamalara sahiptir.

Bu sorunu çözenin birçok olası yolu vardır; fark, kullandığımız veri yapılarında (verileri nasıl temsil ettiğimiz) ve çözüme ulaşmamız için gereken zamandır. Bu tür yöntemlerden biri greedy algoritmasıdır. 'Greedy' kelimesi, bütün nesnelere aşağıya sıraya koyduğumuz 'Fiyat' sırasına göre sıralamaya çalıştığımız anlamına gelir. Bu durumda iniş zamanını Fiyat olarak kullanırız (uçuş ne kadar erken olursa Fiyat o kadar yüksektir). Greedy algoritmalar bize her zaman doğru cevapları vermez, ancak çoğu bize uygun çözümler sunar.

Açgözlü bir grafik renklendirme algoritması kullanarak örnek bir çözümü inceleyelim. Uçuşlar birbirlerine 15 dakikadan daha yakın ise bir ilişki vardır. 9W2400 uçuşuyla başlıyoruz. EK427 ve SG147 - yani 15 dakikadan daha kısa bir sürede ilgili tüm uçuşlara bağlanıyoruz. Her uçuş için bunu tekrarlıyoruz. Bir bağlantı yoksa 15 dakika içinde başka bir uçuşla yeni bir bağlantı. Bağlantısız uçuşlara, kırmızı renkte olan uçuş koridorlarını atayalım. 9W2400, inmek üzere olduğu gibi uçuş koridoru 1'e atandı. Buna bağlı olmayan tüm uçuşlar aynı koridorda olsun - kırmızı olarak işaretlenmiş koridor 1'deki bir uçuşa bağlı olmadıkları sürece kırmızı renkte. Uçuş sırasında 9W1321, 9W2400'e bağlı değildir, kırmızı olan AI620 uçuşuna bağlı oldukları için kırmızı renklendirilemezler. Bir sonraki uçuş koridoru 2, yeşil diyelim. EK427'ye ve buna bağlı olmayan uçuşlara AI561'e atayın. 9W1321, AI561'e bağlı olduğu gibi yeşil renkli olamaz. Üçüncü uçuş koridorunu sarı, SG147'ye ve dördüncü uçuş koridorunu mavi, 9W1321'e atayın.



Anlaşıyor ki, tüm uçuşların güvenli bir şekilde inmesini sağlamak için 4 koridor yeterlidir.

### Anahtar Kelimeler

Planlama, Grafik Boyama, Sıralama

### İlgili Web Siteleri

[https://en.wikipedia.org/wiki/Graph\\_coloring](https://en.wikipedia.org/wiki/Graph_coloring)  
[https://en.wikipedia.org/wiki/Greedy\\_algorithm](https://en.wikipedia.org/wiki/Greedy_algorithm)