

## ODTÜ Bilgisayar Topluluğu Üniversite Öğrencileri Arası IX. Geleneksel Programlama Yarışması Ön Eleme Soruları

Topluluğumuz, 1997'den bu yana geleneksel olarak düzenlediği programlama yarışması serisine bu sene dokuzuncusunu ekliyor. Programlama yarışması, Ulusal Bilim Olimpiyatları formatında, C ve C++ dilleri üzerinden yapılan ve soruları bilgisayar bilimleri alanının temel problemlerinden ilham alan bir yarışmadır. Yarışmamız, dünyadaki benzerleri arasında (ACM, Tübitak, IOI, vs...)

Linux platformunda düzenlenmiş yarışmaların ilki olma ayrıcalığına sahiptir. Ön eleme sorularının son gönderim tarihi 20 Nisan 2006'tır. Ön katılımcılar arasından bu sorular yoluyla belirlenecek yaklaşık 20 finalist, 30 Nisan 2006 tarihinde ODTÜ Bilgisayar Mühendisliği Bölümü'nde düzenlenecek olan finale çağrılacaktır. Özel ödüllü soruyu en iyi çözen yarışmacı ve final sonucunda ilk üç de-

receyi alan finalistler; ödülleri aynı akşam ODTÜ Kültür ve Kongre Merkezi'nde düzenlenecek olan törende alacaklardır.

Sorular ile ilgili teknik detaylar için web sayfamızı (<http://yarisma.cclub.metu.edu.tr>) takip etmeniz gerekmektedir.

Her türlü sorularınız ve daha ayrıntılı bilgi için [yarisma@cclub.metu.edu.tr](mailto:yarisma@cclub.metu.edu.tr) adresine mail atabilirsiniz.

### GENETİK

AGenetik araştırma merkezinin kansere bulunduğu tedavide hücre bölünmeleri nanoteknolojik bir robot ile kontrol edilmektedir. Robota iki boyutlu düzlemde pozitif tamsayı koordinatlı  $n$  adet organel veriliyor. Organeller o şekilde yerleşmişlerdir ki, doğrusal olan herhangi 3 organel bulunmamaktadır. Bu robot gönderildiği hücredeki  $n$  adet organelin ikisinden geçen bir doğru çiziyor. Robot bu işlemi tamamladığında doğrunun iki tarafında da eşit sayıda organel kalıyor ve bölünme sağlıklı bir biçimde gerçekleşiyor. Sizden istenen ise robotu bu doğruyu çizerek şekilde programlamamız.

#### Girdi (agenetik.gir):

• Girdi dosyası `agenetik.gir`'in ilk satırında organel sayısını ifade eden  $n$  ( $1 < n \leq 50000$ ) bulunacaktır,  $n$  çift bir tamsayıdır.

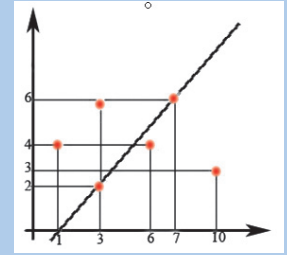
• Takip eden  $n$  satırın herbirisinde iki adet tamsayı bulunacaktır, bu sayılar sıradaki organelin sırasıyla  $x$  ve  $y$  koordinatlarını belirtecektir.

#### Çıktı (agenetik.cik):

• Çıktı dosyası `agenetik.cik`'da 4 adet tamsayı bulunmalıdır. Bu sayılar, verilen organellerden bulunduğunuz doğruyu oluşturan ikisinin koordinatlarını belirtmelidir. Birden fazla çözüm olması durumunda herhangi bir çözümü basabilirsiniz.

#### Örnek:

`agenetik.gir:`  
6  
3 2  
6 4  
7 6  
3 6  
10 3  
1 4  
`agenetik.cik:`  
3 2 7 6



Bulduğumuz doğru (3,2) ve (7,6) koordinatlı organellerden geçmektedir ve düzlemi iki tarafa da eşit sayıda (ikişer) organel kalacak şekilde bölmüştür.

### TELEFON

Üniversitede okuyan Onur okulla ilgili bir çok aktivite düzenlemektedir. Bir aktiviteye karar verildiğinde bir telefon zinciri ile (bir kişi iki veya daha fazla kişiyi arayabilir) aktivite bütün herkese duyurulmaktadır. Günümüzde bir çok telekomünikasyon şirketi bir çok tarife uygulamaktadır. Ayrıca çeşitli şirketlerin çeşitli tarifelerini kullanan üniversite öğrencilerinin her birinin kendine özel bir telefon defteri bulunmakta ve her telefon defterinde en az bir öğrencinin en çok bütün üniversite öğrencilerinin telefon numaraları bulunmaktadır. Toplamda ödenecek telefon ücretini en aza indirecek böyle bir telefon zincirini oluşturmak bazen Onur'un aklını karıştırmaktadır. Bu konuda ona yardımcı olmak için bilgisayar bilimiyle ilgilenen biri olarak size bir program yazmak düşmektedir.

#### Varsayımlar:

- Öğrenci sayısı  $n$ 'dir ( $1 \leq n \leq 5000$ ).
- Telefon numaraları karşılıklı olarak kayıtlıdır, yani eğer Atasay'ın telefon defterinde Yiğit'in telefon numarası varsa, Yiğit'in telefon defterinde de Atasay'ın numarası bulunmaktadır.
- Telefon zincirinin kimden başlayacağı girdiye verilecektir.
- Atasay'ın Yiğit'i aramasının ücreti  $x$  lira ( $x$  bir tamsayıdır,  $0 < x < 100$ ) ise, Yiğit'in Atasay'ı arama ücreti de  $x$ 'dir.

#### Girdi (telefon.gir):

• Girdi dosyası `telefon.gir`'in ilk satırında öğrenci sayısını ifade eden  $n$  tamsayısı verilecektir.

• İkinci satırda  $n$  adet isim (bir isim, içinde boşluk bulunmayan, en fazla 32 karakterden ('a' ile 'z' arasındaki İngilizce karakterler) oluşan bir karakter dizisidir) aralarında birer boşluk bulunacak şekilde verilecektir. Bu isimler okuldaki öğrencilerin isimlerini belirtecektir.

• Takip eden  $n$  satırın her birinde sırasıyla ikinci satırda verilen öğrencilerin telefon defterleri verilecektir. Her satırda ilk olarak rehber sahibinin rehberindeki kişi sayısını ifade eden bir tamsayı bulunacaktır. Daha sonra bu sayı kadar isim ve o isimli kişiyle yapılacak konuşma ücreti verilecektir.

• En son satırda ise telefon zincirinin kimden başlayacağını belirten tek bir isim bulunacaktır.

#### Çıktı (agenetik.cik):

• Çıktı dosyası `agenetik.cik`'da 4 adet tamsayı bulunmalıdır. Bu sayılar, verilen organellerden bulunduğunuz doğruyu oluşturan ikisinin koordinatlarını belirtmelidir. Birden fazla çözüm olması durumunda herhangi bir çözümü basabilirsiniz.

#### Çıktı (telefon.cik):

- Programınız `telefon.cik` dosyasının ilk satırına toplam telefon ücretini basmalısınız.
- İkinci satıra toplam görüşme sayısını basmalısınız.

- Daha sonra telefon zincirini basmalısınız. Zinciri basarken, her satırda bir görüşme bilgisini ifade eden ve aralarında bir boşluk bulunan iki isim yer almalıdır. İlk isim arayan kişiyi, ikinci isim aranan kişiyi belirtecektir. Zincirdeki görüşmelerin sırası şu şartı sağlamalıdır:

Kendisine henüz haber gelmemiş bir kişi (önceki satırlarda aranan kişi olarak belirtilmemiş bir kişi) başkasına haber veremez (dolayısıyla ilk satırda zinciri başlatan kişi, arayan kişi olacaktır).

#### Örnek:

`telefon.gir:`  
7  
onur yigit atasay  
mustafa eda gokdeniz  
cigdem  
2 atasay 3 yigit 7  
3 onur 7 atasay 1 mustafa 4  
3 onur 3 yigit 1 mustafa 5  
4 atasay 5 yigit 4 gokdeniz 13 eda 6  
1 mustafa 6  
2 mustafa 13 cigdem  
3  
1 gokdeniz 3  
onur

`telefon.cik:`  
30  
6  
onur atasay  
atasay yigit  
yigit mustafa  
mustafa gokdeniz  
gokdeniz cigdem  
mustafa eda

### GEZİNTİ

Dünya turu atmak istiyorsunuz fakat çok fazla gezecek paranız yok. Bu yüzden elinizdeki parayla en güzel turu nasıl yapacağınızı bulmanız gerekiyor.

Turu ayarlayacak şirketin ücretlendirmesi şu şekilde yapılıyor:

- Her şehrin bir tur fiyatı vardır.
- Tura başlayacağınız şehir için şehrin tur fiyatı kadar, daha sonraki şehirler içinse kendisi de dahil o şehre varana kadar gezeceğiniz şehirlerden tur fiyatı maksimum olan kadar para ödeyeceksiniz.

Elinizdeki parayı hesapladıktan sonra  $k$  adet şehir gezmeye karar veriyorsunuz. Sizden istenen gezdiğiniz yerlerin ücretiyle ödediğiniz ücret arasındaki ilişki maksimum karlı çıkmak, yani ( $<\text{ödediğiniz tutar}> / <\text{gezdığınız yerlerin toplam tutarı}>$ ) değerini minimize etmek.

©Ali Galip Bayrak - ODTÜ Bilgisayar Topluluğu

## Girdi (gezinti.gir):

• Girdi dosyası gezinti.gir'in ilk satırında, gezmek istediğiniz şehir sayısını ifade eden  $k$  ( $1 \leq k \leq 5$ ) tamsayısı verilecektir.

• Takip eden satırda turu ayarlayacak şirketin gezme imkanı sunduğu şehir sayısını ifade eden  $n$  ( $1 \leq n \leq 30$ ) tamsayısı bulunacaktır.

• Takip eden satırda  $n$  adet tamsayı bulunacaktır. Bu sayılar sırasıyla 1 numaralı şehirden  $n$  numaralı şehre kadar bütün şehirlerin tur fiyatlarını belirtecektir (tur fiyatları 1 ile 200 arasında bir tamsayı olacaktır).

• Bütün şehirler arasında ulaşım olmayacağı için takip eden satırlarda arasında yol bulunan

şehirler verilecektir. Önce yol sayısını ifade eden  $m$  tamsayısı, takip eden  $m$  satırda ise arasında yol bulunan ikişer şehir verilecektir. Yollar çift yönlüdür.

## Çıktı (gezinti.cik):

• Çıktı dosyası gezinti.cik'a gizeceğiniz şehirlerin numaralarını sırayla basmanız gerekmektedir.

Not: Aynı şehri birden fazla kez gezmemeniz gerekmektedir. Gezintiye başlamadan önce bulunduğunuz şehir verilen şehirlerden birisi değildir. Birden fazla çözüm olma durumunda birisini basmanız yeterlidir.

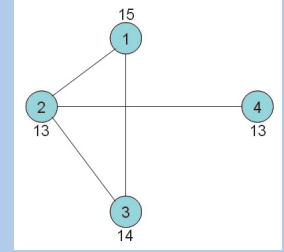
## Örnek:

gezinti.gir:

3  
4  
15 13 14 13  
4  
2 4  
2 1  
1 3  
3 2

gezinti.cik: 4 2 1

Seçtiğimiz yol için değer:  $(13+13+15)/(13+13+15)=1$ . Eğer yolumuzu "1 2 3" olarak seçmiş olsaydık:  $(15+15+15)/(15+13+14)=45/42=1.07$  olacaktı.



## CİN - ÖZEL SORU

Sefalet içinde bir hayat süren kahramanlarımız Atasay ve Ali Galip bir gün Büyük Cin'e ait eski bir kağıt parçası bulurlar. Kağıtta Büyük Cin, Cadı Sila tarafından Ege denizindeki Oğuz adasında bir labirente hapsedileceğini, kendisini bulup kurtaran kişinin 3 dileğini yerine getireceğini yazmıştır. Bunun üzerine adaya bir kayıkla ulaşan ikimiz, adaya vardıklarında Cadı Sila'nın düşürdüğü labirent haritasını bulurlar.

Dikdörtgen şeklindeki harita kare odalardan oluşmaktadır. Boş odalar koridorları, dolu odalar duvarları oluşturmaktadır. 2 girişi olan labirentin bir odasında Büyük Cin hapsedilmiş durumdadır. Labirentteki bazı odalarda bir veya birden fazla hız iksiri bulunmaktadır. Bu iksirler içildikleri zaman için kişiye 2 kat hızlı hareket etme şansı tanımaktadır (kişi iksir için hareket etmeyi seçtiğinde aynı yönde 2 kare gitmek zorundadır). Ayrıca kahramanlarımız istediklerinde hareket etmek yerine çevrelerindeki odalardan birini sonsuza dek kapatabilirler (kişi iksir için oda kapatmayı seçtiği zaman etrafındaki odalardan ikisini kapatmak zorundadır). Bir kişi elinde iksir olduğu halde iksir kullanmadan hareket ederse (veya oda kapatırsa) elindeki tüm iksirleri kaybeder.

Giriş yapacakları kapı konusunda anlaşamayan kahramanlarımız kavga edip farklı kapılardan giriş yapmaya karar verirler. Büyük Cin'in olduğu odaya ilk ulaşan kahramanımız onu kurtarmış olacaktır.

### Varsayımlar:

• Hikayemiz 2 oyuncu arasındaki bir oyun şeklinde oynanacaktır. Sizden istenen bu oyunu oynayacak bir kod yazmanız.

• 1. oyuncu labirentin Kuzey Batı köşesinden, 2. oyuncu ise Güney Doğu köşesinden başlayacaktır.

• Labirentin boyutları  $n \times m$ 'dir.  $n$  satır sayısı,  $m$  sütun sayısını ifade etmektedir ( $2 \leq n, m \leq 30$ ).

• Bir oyuncunun diğerinin bulunduğu odaya gitmesinde (iki oyuncunun aynı anda aynı odada bulunmasında) bir sakınca yoktur.

• Programınız ilk olarak "cin.gir" isimli dosyadan labirent bilgisini okumalıdır. Daha sonra standart girdiden (std.in) oyuncu numarasını okuyarak oyuna başlamalıdır. Oyun esnasında, sıra kendisinde ise standart çıktıya (std.out) hamlesini

basmalı, sıra rakipte ise standart girdiden rakibin hamlesini okumalıdır.

## Girdi-Çıktı:

### cin.gir:

Girdi dosyası cin.gir'in ilk satırında labirentin boyutlarını gösteren  $n$  (satır sayısı) ve  $m$  (sütun sayısı) tamsayıları bulunacaktır. Takip eden  $n$  adet satırın her birinde aralarında birer boşluk bulunan  $m$  adet tamsayı (0, 1, 2 veya 10'dan büyük bir sayı, sırayla bütün odalar için, boş odalar için 0, dolu odalar için 1, Büyük Cin'in bulunduğu oda için 2, iksir bulunan odalar için de odadaki iksir sayısı + 10 sayısı {örn: 3 iksir bulunan bir oda için 13}) bulunacaktır.

### standart girdi - standart çıktı:

Programınız ilk olarak standart girdiden sizin programınızın oyuncu numarasını belirten bir adet tamsayı okuyacaktır. Bu sayı birinci oyuncu için 1, ikinci oyuncu için 2 olacaktır.

İlerleyen aşamalarda, hamle sırası karşındaki oyuncuda ise standart girdiden onun yaptığı hamleyi okuyacak, sıra kendisinde ise standart çıktıya kendi hamlesini yazacaktır. Hamle, hareket için 'H' veya oda kapatmak için 'O', ve yön belirten bir veya 2 karakterden (iksir kullanılıyorsa 2 değilse 1) ve takip eden satır sonu karakterlerinden ('n') oluşmalıdır. (Yön karakterleri: Kuzey: 'K', Güney: 'G', Doğu: 'D', Batı: 'B') (karakterler arasında boşluk yoktur)

### Örnek Hamleler:

HK (kuzey yönünde 1 kare ilerle)

HKK (kuzey yönünde 2 kare ilerle <iksir kullan>)

OK (kuzey yönündeki odayı kapat)

OKG (kuzey ve güney yönlerindeki odaları kapat <iksir kullan>)

### Örnek Hamleler:

HK (kuzey yönünde 1 kare ilerle)

HKK (kuzey yönünde 2 kare ilerle <iksir kullan>)

OK (kuzey yönündeki odayı kapat)

OKG (kuzey ve güney yönlerindeki odaları kapat <iksir kullan>)

### Örnek Oyun:

|    |   |   |   |    |    |
|----|---|---|---|----|----|
| A  | 1 | 0 | 1 | 17 | 1  |
| 17 | 0 | 0 | 0 | 1  | 15 |
| 1  | 1 | 0 | 0 | 0  | 0  |
| 1  | 0 | 0 | 1 | 1  | 14 |
| 1  | 1 | 0 | 0 | 1  | 0  |
| 0  | 2 | 0 | 1 | 1  | B  |

### cin.gir

6 6  
0101171  
17000115  
110000  
1100114  
110010  
020110

|    |   |   |   |    |    |
|----|---|---|---|----|----|
| A  | 1 | 0 | 1 | 17 | 1  |
| 17 | 0 | 0 | 0 | 1  | 15 |
| 1  | 1 | 0 | 0 | 0  | 0  |
| 1  | 0 | 0 | 1 | 1  | 14 |
| 1  | 1 | 0 | 0 | 1  | 0  |
| 0  | 2 | 0 | 1 | 1  | B  |

A hamle: HG

|   |   |   |   |    |    |
|---|---|---|---|----|----|
| 0 | 1 | 0 | 1 | 17 | 1  |
| A | 0 | 0 | 1 | 15 |    |
| 1 | 1 | 0 | 0 | 0  | 0  |
| 1 | 0 | 0 | 1 | 1  | 14 |
| 1 | 1 | 0 | 0 | 1  | 0  |
| 0 | 2 | 0 | 1 | 1  | B  |

A 7 iksir aldi.

B hamle: HK

|   |   |   |   |    |    |
|---|---|---|---|----|----|
| 0 | 1 | 0 | 1 | 17 | 1  |
| A | 0 | 0 | 1 | 15 |    |
| 1 | 1 | 0 | 0 | 0  | 0  |
| 1 | 0 | 0 | 1 | 1  | 14 |
| 1 | 1 | 0 | 0 | 1  | 0  |
| 0 | 2 | 0 | 1 | 1  | B  |

A hamle: HDD (iksir kullan)

|   |   |   |   |    |    |
|---|---|---|---|----|----|
| 0 | 1 | 0 | 1 | 17 | 1  |
| 0 | 0 | A | 0 | 1  | 15 |
| 1 | 1 | 0 | 0 | 0  | 0  |
| 1 | 0 | 0 | 1 | 1  | 14 |
| 1 | 1 | 0 | 0 | 1  | 0  |
| 0 | 2 | 0 | 1 | 1  | B  |

A nun 6 iksirini kaldi.

B hamle: HK

|   |   |   |   |    |    |
|---|---|---|---|----|----|
| 0 | 1 | 0 | 1 | 17 | 1  |
| 0 | 0 | A | 0 | 1  | 15 |
| 1 | 1 | 0 | 0 | 0  | 0  |
| 1 | 0 | 0 | 1 | 1  | B  |
| 1 | 1 | 0 | 0 | 1  | 0  |
| 0 | 2 | 0 | 1 | 1  | 0  |

B 4 iksir aldi.

A hamle: HGG

|   |   |   |   |    |   |
|---|---|---|---|----|---|
| 0 | 1 | 0 | 1 | 17 | 1 |
| 0 | 0 | 0 | 1 | 15 |   |
| 1 | 1 | 0 | 0 | 0  | 0 |
| 1 | 0 | A | 1 | 1  | B |
| 1 | 1 | 0 | 0 | 1  | 0 |
| 0 | 2 | 0 | 1 | 1  | 0 |

A nun 5 iksirini kaldi.

B hamle: HK

|   |   |   |   |    |   |
|---|---|---|---|----|---|
| 0 | 1 | 0 | 1 | 17 | 1 |
| 0 | 0 | 0 | 1 | 15 |   |
| 1 | 1 | 0 | 0 | 0  | 0 |
| 1 | 0 | A | 1 | 1  | B |
| 1 | 1 | 0 | 0 | 1  | 0 |
| 0 | 2 | 0 | 1 | 1  | 0 |

B iksir kullanmadığı için 0 iksirini kaldi.

A hamle: OKB

|   |   |   |   |    |   |
|---|---|---|---|----|---|
| 0 | 1 | 0 | 1 | 17 | 1 |
| 0 | 0 | 0 | 1 | 15 |   |
| 1 | 1 | 0 | 0 | 0  | 0 |
| 1 | 1 | 0 | 1 | 1  | 0 |
| 1 | 1 | 0 | 0 | 1  | 0 |
| 0 | 2 | 0 | 1 | 1  | 0 |

A nun 4 iksirini kaldi.

B hamle: HB

|   |   |   |   |    |   |
|---|---|---|---|----|---|
| 0 | 1 | 0 | 1 | 17 | 1 |
| 0 | 0 | 0 | 1 | 15 |   |
| 1 | 1 | 0 | 0 | 0  | 0 |
| 1 | 1 | 0 | 1 | 1  | 0 |
| 1 | 1 | 0 | 0 | 1  | 0 |
| 0 | 2 | 0 | 1 | 1  | 0 |

A hamle: HGG

|   |   |   |   |    |   |
|---|---|---|---|----|---|
| 0 | 1 | 0 | 1 | 17 | 1 |
| 0 | 0 | 0 | 1 | 15 |   |
| 1 | 1 | 0 | 0 | 0  | 0 |
| 1 | 1 | A | 1 | 1  | 0 |
| 1 | 1 | 0 | 0 | 1  | 0 |
| 0 | 2 | 0 | 1 | 1  | 0 |

A nun 3 iksirini kaldi.

B hamle: HB

|   |   |   |   |    |   |
|---|---|---|---|----|---|
| 0 | 1 | 0 | 1 | 17 | 1 |
| 0 | 0 | 0 | 1 | 15 |   |
| 1 | 1 | 1 | B | 0  | 0 |
| 1 | 1 | 0 | 1 | 1  | 0 |
| 1 | 1 | 0 | 0 | 1  | 0 |
| 0 | 2 | A | 1 | 1  | 0 |

A hamle: OK

|   |   |   |   |    |   |
|---|---|---|---|----|---|
| 0 | 1 | 0 | 1 | 17 | 1 |
| 0 | 0 | 0 | 1 | 15 |   |
| 1 | 1 | 1 | B | 0  | 0 |
| 1 | 1 | 0 | 1 | 1  | 0 |
| 1 | 1 | 0 | 0 | 1  | 0 |
| 0 | 2 | A | 1 | 1  | 0 |

A iksir kullanmadığı için elindeki iksirleri kaybetti.

B hamle: HK

|   |   |   |   |    |    |
|---|---|---|---|----|----|
| 0 | 1 | 0 | 1 | 17 | 1  |
| 0 | 0 | 0 | B | 1  | 15 |
| 1 | 1 | 1 | 0 | 0  | 0  |
| 1 | 1 | 0 | 1 | 1  | 0  |
| 1 | 1 | 0 | 0 | 1  | 0  |
| 0 | 2 | A | 1 | 1  | 0  |

A hamle: HB

|   |   |   |   |    |    |
|---|---|---|---|----|----|
| 0 | 1 | 0 | 1 | 17 | 1  |
| 0 | 0 | 0 | B | 1  | 15 |
| 1 | 1 | 1 | 0 | 0  | 0  |
| 1 | 1 | 0 | 1 | 1  | 0  |
| 1 | 1 | 0 | 0 | 1  | 0  |
| 0 | 2 | A | 1 | 1  | 0  |

1. oyuncu oyunu kazandı