



ODTÜ Bilgisayar Topluluğu

7. ÜNİVERSİTE ÖĞRENCİLERİ ARASI PROGRAMLAMA YARIŞMASI ÖN ELEME SORULARI

Sorularla ilgili teknik detaylar için web sayfamıza

(<http://yarisma.cclub.metu.edu.tr>) bakmanız gerekmektedir. Her türlü sorunuz ve daha ayrıntılı bilgi için yarisma@ccclub.metu.edu.tr adresine mail atabilirsiniz.

Bayan Cansın'ın Yolculuğu

Uzun bir seyahate çıkacak olan Bayan Cansın, arabasının 45 litrelik deposunu doldurur ve elindeki notları incelemeye başlar. Bayan Cansın'ın notlarındaki bilgiler, şehirleri, doğrudan yol ile bağlı şehir ikilileri arasında arabasının harcayacağı benzin miktarını ve bir şehirde eğer benzinlik varsa Bayan Cansın'ın bu şehirden alabileceği maksimum benzin miktarını içermektedir. Bayan Cansın elindeki bu notları kullanarak, bulunduğu şehirden gitmek istediği şehre yolda kalmadan (yol üzerinde seyahat halindeyken deposundaki benzin miktarı 0'a düşmeden) en az miktarda benzin harcayarak ulaşmak istiyor. Sizden istediğimiz Bayan Cansın'a yardımcı olacak bir program yazmanız.

VARSAYIMLAR

- Şehirler 1'den n'e kadar ardışık tamsayılar ile gösterilmektedir. Bayan Cansın 1 numaralı şehirde bulunmaktadır ve n numaralı şehre ulaşmak istemektedir..
- Maksimum x litre benzin alınabilecek bir şehirden Bayan Cansın isteği doğrultusunda ve deposunun kapasitesini aşmayacak şekilde y litre benzin alabilir. ($1 \leq y \leq x$)
- Yollar çift yönlü olup, her iki yön için de harcanacak benzin miktarları aynıdır.

- Programınız "yolculuk.gir" isimli dosyadan girdileri alıp "yolculuk.cik" isimli dosyaya çıktıları yazmalıdır.

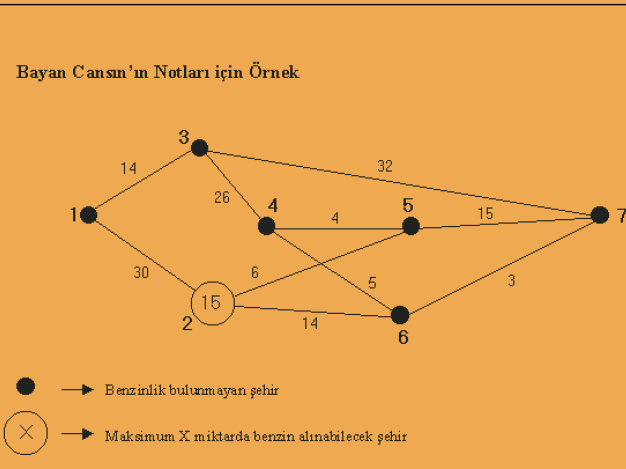
GİRDİ (yolculuk.gir)

- İlk satır, sırasıyla şehir sayısı (n), şehirler arasındaki mevcut yol sayısı (m) ve benzinlik bulunan şehir sayısını (b) ifade eden, aralarında tam olarak bir boşluk bulunan 3 adet tamsayıdan oluşacaktır. ($2 \leq n \leq 500$)
- Takip eden 'b' satırda sırasıyla şehir numarası (u) ve alınabilecek maksimum benzin miktarını (v) gösteren ve aralarında tam olarak bir boşluk bulunan 2 tamsayı oluşacaktır. ($1 \leq v \leq 45$)
- Bundan sonraki 'm' satırın her birisinde ise aralarında tam olarak bir boşluk bulunan 3 adet tamsayı; p, r ve s ($1 \leq s \leq 45$) yer alacak ve bu sayıların anlamı "p şehri ile r şehri arasındaki yolculuk için s litre benzin harcanır" olacaktır.

ÇIKTI (yolculuk.cik)

- Bu dosya Bayan Cansın'ın yolculuğu için harcanabilecek minimum benzin miktarını ifade eden tek bir tamsayıdan oluşacaktır.

ÖRNEK



yolculuk.gir :
7 10 1
2 15
1 3 14
1 2 30
3 4 26
3 7 32
4 5 4
5 7 15
7 6 3
6 2 14
5 2 6
4 6 5

yolculuk.cik :
47

Bilye

Eski çağlarda Zekado ülkesinde şöyle bir oyun oynanmış:

- Ortaya belli sayıda (n) bilye konur.
- İki kişi sıra ile hamleler yapar.
- Birinci oyuncu ilk hamlesinde ortadan 2 ya da 3 bilye alır.
- Sırası gelen oyuncu, bir önceki oyuncunun aldığı bilye sayısı k ise, $k+1 \leq m \leq 2k$ olacak şekilde m tane bilye alır. Son hamlede geriye kalan bilyelerin sayısı $\leq k$ ise bu bilyelerin hepsini alabilir.
- Son bilye grubunu alan oyunu kazanır.

Örnek bir oyun:

Ortada 25 tane bilye var:

	Aldığı Bilye Sayısı	Kalan Bilye Sayısı
1. Oyuncu	2	23
2. Oyuncu	3	20
1. Oyuncu	5	15
2. Oyuncu	8	7
1. Oyuncu	7	0

Bu durumda son hamleyi yapmış olan 1. oyuncu kazanıyor.

Sizden istenen ise bu oyunu hem 1. hem de 2. oyuncu için oynayan bir program yazmanız.

VARSAYIMLAR

- $2 \leq n \leq 100$
- Programınız standart girdiden (stdin) hamleleri alıp standart çıktıya (stdout) hamlelerini (aldığı bilye sayılarını) yazmalıdır.
- GİRDİ - ÇIKTI
- Programınız oyun başlarken standart girdiden bilye sayısını (n) ve oyuncu numarasını belirten iki adet tamsayı okuyacaktır. Oyuncu numarasının değeri, birinci oyuncu iseniz 1, ikinci oyuncu iseniz 2 olacaktır.
- İlerleyen aşamalarda, hamle sırası karşındaki oyuncu da ise onun yaptığı hamleyi okuyacak (1 adet tamsayı), sıra kendinde ise hamlesini yazacaktır (1 adet tamsayı).

DEĞERLENDİRME

- Verilen 'n' değerleri için, herhangi bir kod diğer bütün kodlarla hem 1. hem de 2. oyuncu için oyunu oynayacaktır.
- Herhangi bir anda yanlış bir hamle yapan oyuncu o oyunu kaybetmiş sayılacaktır.

Süpzek Piramitlerde

Kahramanımız Süpzek eski Mısır uygarlığı üzerine araştırmalar yapmak için girdiği Keops Piramiti'nde bir kutu bulur. Kutunun içinde satranç tah-tası şeklinde ve tüm karelerinde harfler yazılı iki adet taş vardır. Kutuyu bi-raz daha karıştırınca içerisinde bir de şifrelenmiş not bulur. Yetenekli kah-ramanımız biraz uğraştıktan sonra notu çözer:

“en büyük ortak kareyi bul, anahtar orada!”.

Nottaki mesaja bir anlam veremeyen Süpzek etrafını daha dikkatli ince-lediğinde bulduğu kutunun altında başka bir kutu daha olduğunu farkedir. Kutuyu açmaya çalışsa da başarılı olamaz, sonunda notta yazan şeyi yapma-ya karar verir. Bakalım siz ona yardımcı olabilecek misiniz?

VARSAYIMLAR

- Karelerin boyu aynı ve 'n'dir ($2 \leq n \leq 150$)
- Programınız “piramit.gir” isimli dosyadan girdileri alıp “piramit.cik” isimli dosyaya çıktılarını yazmalıdır.

GİRDİ (piramit.gir)

Girdi dosyası piramit.gir'in ilk satırında karelerin boyutunu gösteren n tamsayısı bulunacaktır. Takip eden $2 \times n$ adet satırın her birinde aralarında birer boşluk bulunan n adet karakter (İngiliz alfabesindeki büyük harfler) bulunacaktır. İlk n adet satır birinci taş üzerindeki kareleri, sonraki n adet satır ise ikinci taş üzerindeki kareleri temsil etmektedir.

ÇIKTI (piramit.cik)

Çıktı dosyası piramit.cik'in ilk satırında en büyük ortak karenin boyutu-

piramit.gir:

```
10
A A F S F S F S F S
B B T T T T T T T T
C C R R T T R R T T
B B B B N M N M N P
A A A A Q N Q Q Q M
B B B B N Q M M M M
T T B A X X X M N M
S F S F W W W Q Q Q
R R R R Z Z W Q Q Z
C C R C Z Q X M N Z
L L L L Y Y Y Y L Y
Q N Q Q Q M K K L L
N Q M M M M L K Y
X X X M N M Y Y K L
W W W Q Q A A A A
Z Z W Q Q Z Y K L A
Z Q X M N Z C C R R
O O P O P O P O P O
A A F S F S F S F S
B B T T T T T T T T
```

piramit.cik:

```
6
5 5
2 1
```

1. kare

A	A	F	S	F	S	F	S	F	S
B	B	T	T	T	T	T	T	T	T
C	C	R	R	T	T	R	R	T	T
B	B	B	B	N	M	N	M	N	P
A	A	A	A	Q	N	Q	Q	Q	M
B	B	B	B	N	Q	M	M	M	M
T	T	B	A	X	X	X	M	N	M
S	F	S	F	W	W	W	Q	Q	Q
R	R	R	R	Z	Z	W	Q	Q	Z
C	C	R	C	Z	Q	X	M	N	Z

2. kare

L	L	L	L	Y	Y	Y	Y	L	Y
Q	N	Q	Q	Q	M	K	K	L	L
N	Q	M	M	M	M	L	L	K	Y
X	X	X	M	N	M	Y	Y	K	L
W	W	W	Q	Q	Q	A	A	A	A
Z	Z	W	Q	Q	Z	Y	K	L	A
Z	Q	X	M	N	Z	C	C	R	R
O	O	P	O	P	O	P	O	P	O
A	A	F	S	F	S	F	S	F	S
B	B	T	T	T	T	T	T	T	T

nu gösteren bir adet tamsayı bulunmalıdır. İkinci ve üçüncü satırlarda ise, en büyük ortak karenin sırasıyla birinci ve ikinci taş üzerinde bulunduğu ye-rin sol üst koordinatını satır ve sütun olarak gösteren ik adet tamsayı bu-lunmalıdır. Bir taşın sol üst köşesinin koordinatı (1, 1), sağ alt köşesinin ko-ordinatı ise (n,n)'dir.

Kamyonlar Kavun Taşır

=Özel Soru=

Zekado ülkesinin kralı VII. Süpzek, ülkesinde kavun yetiştirilen n güney şehirden kuzeydeki n şehre kavun taşımak ister. Bu iş için n güney şeh-rin her birine birer tane kamyon gönderir. Bu kamyonlar gönderildikleri şehirlerden yola çıka-rak kuzey şehirlere doğru yol alır (her kamyonun gideceği şehir baştan bellidir). Kamyonlar bir yo-lu tek başına giderken her km'de a litre mazot harcayorsa, m kamyon birlikte giderlerken her km'de toplam $a \cdot \log_2(m+1)$ litre mazot harcarlar. Amacımız Kral Süpzek'e yardım etmek ve kam-yonlarının mümkün olduğunca az mazot harcaya-rak, n kaynak şehirden n hedef şehre kavunları ta-şımalarını sağlamaktır.

VARSAYIMLAR

- $2 \leq n \leq 20$.
- Ülkede toplam 'm' adet şehir var-dır ($4 \leq m \leq 50$).
- Yollar çift yönlüdür. (b'den c'ye yol varsa, c'den de b'ye yol vardır)
- 1 birim zaman, aralarında yol bulunan her-hangi iki şehir arasını herhangi bir kamyonun ka-tetmesi için geçen zamandır (bütün şehir ikilileri ve bütün kamyonlar için aynı).

- Kamyonların şehirlerden ayrılma zamanları birim zaman cinsinden hesaplanır.
- Eğer bir kamyon önce b şehirden, daha sonra c şehirden geçiyorsa, b şehirden ayrılma zamanı c şehirden ayrılma zamanından küçük ol-malıdır.
- Bir kamyon herhangi bir şehirde istediği ka-dar bekleyebilir.
- Programınız “kamyon.gir” isimli dosyadan girdileri alıp “kamyon.cik” isimli dosyaya çıktıları-nı yazmalıdır.

GİRDİ (kamyon.gir)

- İlk satırda, toplam şehir sayısı (m) verilecek-tir. (Bir adet tamsayı)
- İkinci satırda kaynak/hedef şehirlerin sayı-sı (n) verilecektir. (Bir adet tamsayı)
- Üçüncü satırda aralarında tam olarak birer boşluk bulunan n tane tamsayı bulunacaktır. Bu sayılar kaynak şehirlerin numarasını belirtecek-tir. (Şehirler 1'den m'e kadar numaralandırılmıştır).
- Dördüncü satırda aralarında tam olarak bi-rer boşluk bulunan n tane tamsayı bulunacaktır. Bu sayılar hedef şehirlerin numarasını belirtecek-tir. (k'ıncı kamyon, üçüncü satırdaki k'ıncı sırada bulunan şehirden, dördüncü satırda k'ıncı sırada bulunan şehre gidecektir.)
- Beşinci satırda ülkedeki toplam yol sayısı (t) verilecektir. (Bir adet tamsayı)
- Takip eden t satırda, her satırda aralarında tam olarak birer boşluk bulunan 3 tane tamsayı bulunacaktır. Bu sayıların ilk ikisi aralarında yol bulunan şehirlerin numarasını, sonuncusu ise bu yolun uzunluğunu belirtecektir.

ÇIKTI (kamyon.cik)

- İlk satırda harcanan mazot miktarı verilme-lidir (a=1 kabul edilmelidir, sonuç ondalıklı olarak ve noktadan sonraki kısmı 2 haneli olacak şekilde basılmalıdır).

- Takip eden n satır için:

$k+1$ 'inci satırda, ilk olarak, k numaralı kamy-onun izlediği yol (kaynak şehirden hedef eyalate ulaşınca kadar sıra ile geçtiği şehirlerin numa-raları, aralarında birer boşlukla), daha sonra son şehir hariç bu şehirlerden ne zaman yola çıktığı aralarında birer boşlukla verilmelidir.

DEĞERLENDİRME

Kodlarınızın en az mazot harcama ve en hızlı çö-zümü üretme kriterlerinden oluşan bir puanlandır-ma sistemiyle değerlendirilecektir.

ÖRNEK

kamyon.gir:

```
6
2
1 3
2 5
8
1 4 2
1 3 4
3 4 3
1 6 4
3 6 5
4 6 3
6 2 4
6 5 5
```

kamyon.cik:

```
İki adet çıktı örneği:
18.74
1 4 6 2 0 1 2
3 4 6 5 0 1 2
20.00
1 4 6 2 0 1 2
3 4 6 5 0 2 3
```

Açıklamalar:

Sorular ile ilgili teknik detaylar için web sayfamıza (<http://yarisma.cclub.metu.edu.tr>) bakmanız gerekmektedir. Her türlü sorunuz ve daha ayrıntılı bilgi için yarisma@cclub.metu.edu.tr adresine mail atabilirsiniz.