

Mantıksal İfadeler

Mantıksal bir ifade, bu ifadede geçen önermeler ve bu önermelerin değerleri veriliyor ve bu ifadenin sonucu isteniyor. Bir ifade, bizim tanımımızda:

- Bir önerme (tek bir küçük harfle gösterilsin, örn: p)
 - Bir ifadenin “değil”i (ifadenin başında “!” işareti ile gösterilsin)
 - İki ifadenin “ve”lenmiş hali (ifadelerin arasına “&” işareti koyularak gösterilsin)
 - İki ifadenin “veya”lanmış hali (ifadelerin arasına “|” işareti koyularak gösterilsin)
- olabilir.

Varsayımlar

- n adet önerme vardır ($1 \leq n \leq 100$).
- Parantezin önceliği “değil”e göre, “değil”in önceliği “ve”ye göre, “ve”nin önceliği “veya”ya göre daha büyüktür.

Girdi

- Girdiler “mantik.gir” isimli dosyadan okunacaktır.
- İlk satırda önermelerin sayısını ifade eden n verilecektir.
- Takip eden satırda n adet harf, önermelerin isimlerini belirtecektir.
- Takip eden satırda önermelerin değeri

rini ifade eden n adet harf verilecektir. Bu harflerden her birisi D (doğru) veya Y (yanlış)dir.

- Takip eden satırda değerini bulmamız istenen ifade verilecektir.

Çıktı

- Çıktılar “mantik.cik” isimli dosyaya yazılacaktır.
- Tek bir harf bulunacaktır. İfade doğru ise D, yanlış ise Y basılacaktır.

Örnek

mantik.gir:

3

p q r

Y D Y

!(p | q & !(p | r))

mantik.cik:

Y

İfadenin öncelik sırasını göz önünde bulundurup tamamen parantezli hale getirirsek:

!(p | (q & (!(p | r)))) =

!(p | (q & (!(Y | Y)))) =

!(p | (q & (!Y))) =

!(p | (D & D)) =

!(Y | D) =

!(D) =

Y

Doğruluk Tablosu

	p ve q (p & q)	p veya q (p q)	değil p (!p)
p = Y, q = Y	Y	Y	D
p = Y, q = D	Y	D	D
p = D, q = Y	Y	D	Y
p = D, q = D	D	D	Y

Mantıksal İfadeler 2

Bu kez mantıksal ifadenin kendisi ve değeri veriliyor ve önermelerin alabileceği olası bütün değerler dizisi isteniyor.

Varsayımlar

- n adet önerme vardır ($1 \leq n \leq 20$).

Girdi

- Girdiler “mantik2.gir” isimli dosyadan okunacaktır.
- İlk satırda önermelerin sayısını ifade eden n verilecektir.
- Takip eden satırda n adet harf, önermelerin isimlerini belirtecektir.
- Takip eden satırda bir ifade verilecektir.
- Takip eden satırda verilen ifadenin değeri verilecektir.

Örnek

mantik2.gir:

3

p q s

p & q | !p & s

D

mantik2.cik:

D D Y

D D D

Y D D

Y Y D

mantik2.cik dosyasındaki her bir satırdaki değerler sırasıyla p, q ve s'e karşılık gelen değerleri ifade etmektedir.örn: ilk satır için p=D, q=D, s=Y. Girdide verilen ifadede yerine koyup sağlayıp sağlamadığını görebiliriz.

Çıktı

- Çıktılar “mantik2.cik” isimli dosyadan okunacaktır.

- Her satırda olası bir dizilim, girdide verilen sırada verilecektir.

Geçen Sayımızdaki Soruların Çözümleri

Haberleşme

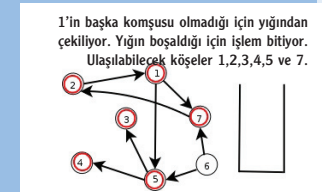
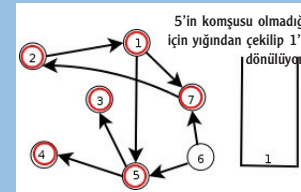
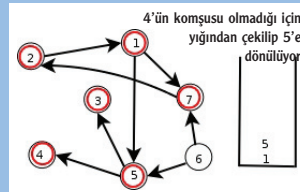
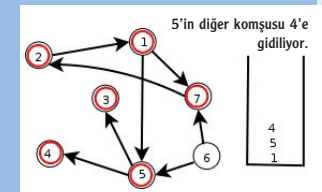
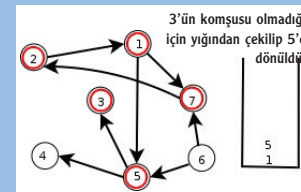
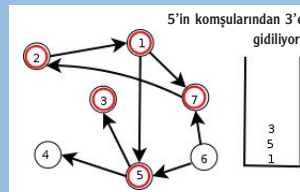
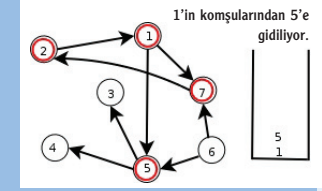
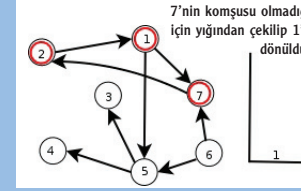
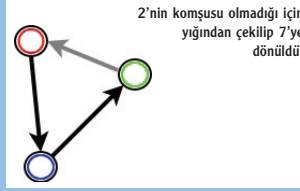
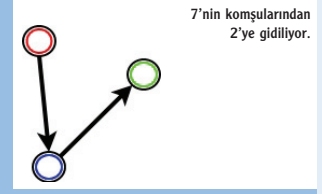
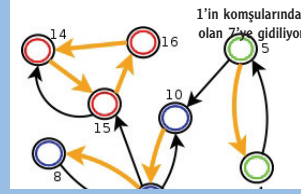
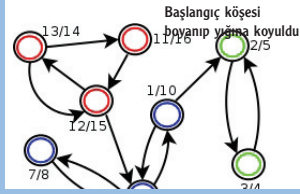
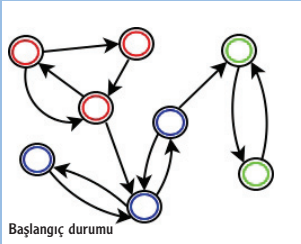
Bu problem bilgisayar biliminde "Depth First Search (DFS)" yani "Derinlik Öncelikli Arama" olarak bilinir. Algoritmayı şu şekilde özetleyebiliriz:

1. Verilen yönlü çizgede⁽¹⁾ bütün köşeler siyaha boyanır ve boş bir yığın⁽²⁾ açılır.

2. Başlangıç köşesi kırmızıya boyanır ve yığına atılır.

3. En son kırmızıya boyanan köşeden gidilebilecek siyah bir köşe seçilir. Bu köşe kırmızıya boyanır ve yığına basılır. Eğer gidilebilecek bir köşe yoksa yığından bu köşe çekilir ve yığın en üstündeki köşeye döndülür.

4. Eğer yığın boşaldıysa, kırmızıya boyanmış köşeler başlangıç köşesinden ulaşılacak köşeleri ifade eder. Yığın boş değilse tekrar 3'e döndülür. Örnek verecek olursak:



(1) Çizge, köşeler ve bu köşeleri birleştiren kenarlardan oluşan yapıdır. Yönlü çizgelerde kenarlar yön belirtir.

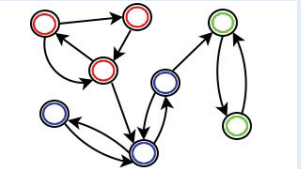
(2) Yığın, şarjör benzeri bir yapıdır. Yığında genelde 3 işlem yaparız: Yığın en üzerine yeni bir eleman basmak, yığın en üzerin-

deki elemanı çekmek ve yığın en üzerindeki elemanı görmek.

Yukardaki örneğimizde yığın ve çizge örnekleri görüyoruz.

Haberleşme 2

Bu problemi çözmek için ilk aşamada verilen yönlü çizgedeki "strongly connected component"leri yani "güçlü bağlı bileşen"leri (GBB) çıkaracağız. Bir GBB'nin özelliği, bütün köşelerinden diğer bütün köşelerine yol olmasıdır. Örnek verecek olursak, aşağıdaki çizgede 3 adet GBB vardır (her GBB farklı renktedir):

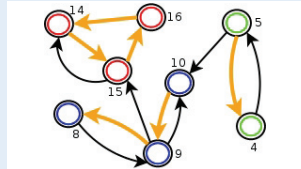
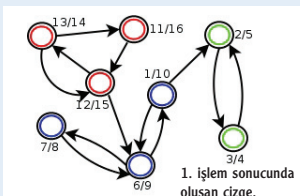


Bir çizgedeki GBB'leri şu şekilde bulabiliriz:

1. Önceki soruda bahsettiğimiz DFS yöntemi kullanılarak her köşenin yığına girme ve çekilme zamanları hesaplanır. Bunu yaparken rastgele bir köşeden başlanır, o köşe için DFS yapılır, bu köşe için DFS bitti ise henüz gezilme-

miş köşelerden birisi alınarak o köşeden DFS yapılır ve bu şekilde bütün köşelerin gezilmesi sağlanır.

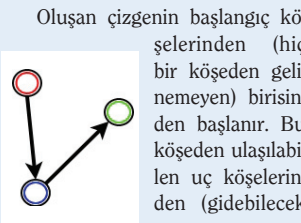
2. Çizgenin tersi alınır (her kenarın yönü ters çevrilir). Bu çizgede de DFS'ler uygulanır. DFS'ler uygulanırken şöyle bir yol izlenir: ilk önce bitiş zamanı en büyük olan köşe için DFS uygulanır, sonra kalan köşelerden en büyük bitiş zamanına sahip olan seçilerek bu köşeden DFS uygulanır ve bu işlem tekrarlanarak bütün köşelerin gezilmesi sağlanır. Her DFS bir bileşeni oluşturur (aynı DFS'de gezilen bütün köşeler aynı bileşendir ve bu bileşen sadece o DFS'de gezilen köşelerden oluşur). Bu aşamada kaç DFS yaptıysak o ka-



2. işlem sonucunda oluşan çizge (DFS sırasında gezilen kenarlar sarı ile gösteriliyor).

dar bileşen (GBB) var demektir.

Sorumuzdaki çizgedeki herhangi bir GBB'yi ele alırsak, bu GBB içerisindeki bir kişiye ulaşan bütün haberler, GBB içerisindeki herkese ulaşır diyebiliriz. Eğer tüm çizgeyi tek bir GBB haline getirebilirsek sorunu çözmüş oluruz. Çizgemizdeki GBB'lerin her birisini büyük bir köşe gibi düşünüp GBB'leri bir çizge haline getirirsek:



komşusu olmayan köşeler) birisi alınarak bu uç köşeden başlangıç köşesine bir kenar eklenir. Bu şekilde oluşan yeni çizgede yeni bir GBB elde ederiz (çünkü, başlangıç köşesinden uç köşesine ulaşılabilir ve uç köşesinden başlangıç köşesine kenar ekleyerek uçtan da başlangıça ulaşılmasını sağladık). Yeni GBB'deki köşeleri birleştirdikten sonra tekrar yeni bir başlangıç ve uç köşesi belirlenerek işlem bir başlangıç ve bir uç köşesi kalana kadar sürdürülür. Son kalan uçtan son kalan başlangıça bir kenar çizilerek işlem tamamlanır.

Örneğimizde şekilde verilen gri kenarı ekleyerek işlemi bitirebiliriz (kırmızı köşe başlangıç köşesi, yeşil köşe uç köşesidir).

Ana örneğimizde yeşil boyalı herhangi bir köşeden kırmızıya boyalı herhangi bir köşeye kenar eklememiz yeterlidir.