

DİKKAT: Çalışma Notu Hakkında

Aşağıda yer alan sorular ve çözümleri tamamen bireysel çalışma ve pratik yapma amacıyla oluşturulmuştur. Bu doküman;

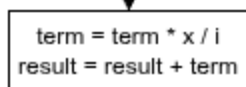
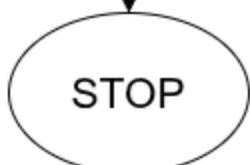
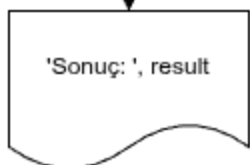
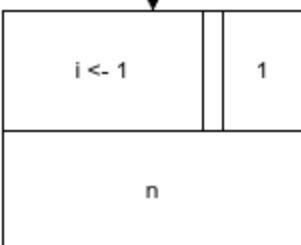
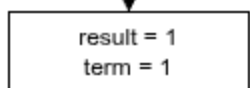
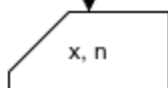
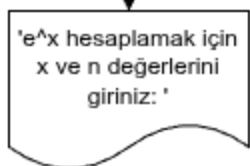
- Yazım hataları, mantıksal yanlışlıklar içerebilir.
- Resmi sınav müfredatıyla %100 uyum garantisi vermez.
- Cevap anahtarı "tek doğru" yolu temsil etmiyor olabilir.

Bu dokümanı kullanırken ana ders kitaplarınızı ve ders notlarınızı teyit mekanizması olarak kullanmanız önemle rica olunur. Oluşabilecek hatalardan hazırlayan kişi sorumlu tutulamaz.



Kullanıcıdan alınan x ve n değerlerine göre e^x ifadesini hesaplayan algoritmanın akış diyagramını çiziniz.

$$e^x = 1 + x + \frac{x^2}{2!} + \dots + \frac{x^n}{n!}$$



START

'Puzzle parça
sayısını giriniz: '

n

n

i <- 1

1

n

'Parça numarasını
giriniz: '

A[i] = 0

x

x != 0

count = 0

i <- 1

1

n

'Toplam ', count, '
adet parça
kaybolmuştur.'

STOP

F

A[i] == 0

T

count = count + 1

i, '. parça
kaybolmuştur.\n'

A[x] = 1

'Parça numarasını
giriniz: '

x

n parçadan oluşan bir yapbozun parçaları 1'den n'e kadar numaralandırılmıştır. Yapbozun bazı parçaları kaybolmuştur. Kullanıcı, öncelikle yapbozun toplam parça sayısını (n), ardından elinde bulunan parçaların numaralarını girecektir. Kullanıcı, elindeki tüm parçaları girdikten sonra 0 değerini girerek işlemi sonlandırır. Kaybolan parçaların numaralarını ve toplam kayıp parça sayısını bulan algoritmanın akış diyagramını çizin. (Kullanıcının tüm verileri doğru girdiği varsayılacaktır.)

START

'Çokgenin kenar sayısını giriniz: '

n

i <- 1

1

n

total = 0

i <- 1

1

n-1

Soru: n kenarlı kapalı bir çokgenin geometrik hesaplamalarını yapan bir algoritma tasarlanacaktır. Kullanıcıdan öncelikle köşe sayısı (n) istenecek, ardından çokgenin köşelerine ait (x, y) koordinatları **ardışık sırayla (saat yönünde veya tersinde)** girilecektir.

Bu verileri kullanarak:

1. Her bir kenarın uzunluğunu ayrı ayrı hesaplayan,
 2. Çokgenin toplam çevre uzunluğunu bulan,
 3. Sonuçları ekrana yazdıran,
- algoritmanın akış diyagramını çiziniz.

i, '. koordinatı
giriniz (x, y): '

X[i], Y[i]

length = SQRT((X[n]-X[1]) * (X[n]-X[1]) + (Y[n]-Y[1]) * (Y[n]-Y[1]))
total = total + length

n, '. kenar
uzunluğu: ', length

'Çevre uzunluğu:
, total

STOP

length = SQRT((X[i+1]-X[i]) * (X[i+1]-X[i]) + (Y[i+1]-Y[i]) * (Y[i+1]-Y[i]))
total = total + length

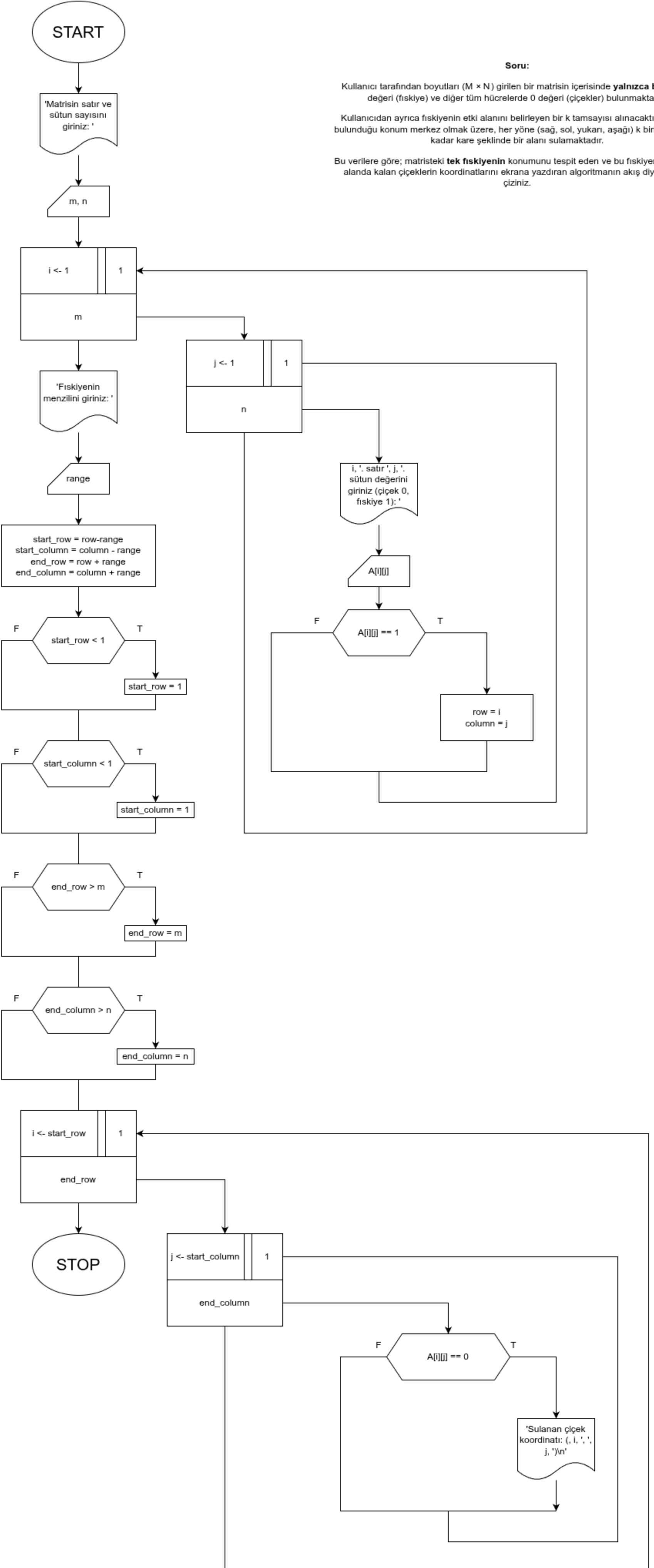
i, '. kenar
uzunluğu: ', length

Soru:

Kullanıcı tarafından boyutları ($M \times N$) girilen bir matrisin içerisinde **yalnızca bir adet 1** değeri (fiskiye) ve diğer tüm hücrelerde 0 değeri (çiçekler) bulunmaktadır.

Kullanıcıdan ayrıca fiskiyenin etki alanını belirleyen bir k tamsayısı alınacaktır. Fiskiye, bulunduğu konum merkez olmak üzere, her yöne (sağ, sol, yukarı, aşağı) k birim uzaklığa kadar kare şeklinde bir alanı sulamaktadır.

Bu verilere göre; matristeki **tek fiskiyenin** konumunu tespit eden ve bu fiskiyenin suladığı alanda kalan çiçeklerin koordinatlarını ekrana yazdıran algoritmanın akış diyagramını çiziniz.



Soru:

Kullanıcıdan şehir sayısı (N) ve bu şehirler arasındaki uçuş bağlantılarını temsil eden bir matris (komşuluk matrisi) alınacaktır.

- Matrisin (i,j) hücresindeki **1** değeri, i. ve j. şehirler arasında uçuş olduğunu; **0** değeri ise uçuş olmadığını gösterir.
- Uçuşlar çift yönlüdür.

Bu bilgilere göre:

1. Hiçbir şehre uçuşu bulunmayan şehirleri (izole şehirler) tespit eden,
 2. En fazla uçuşa sahip olan şehri (birden fazla ise herhangi birini) ve bu uçuş sayısını bulan,
- algoritmanın **akış diyagramını (flowchart)** çiziniz.

Örnek Senaryo:

Girdi:

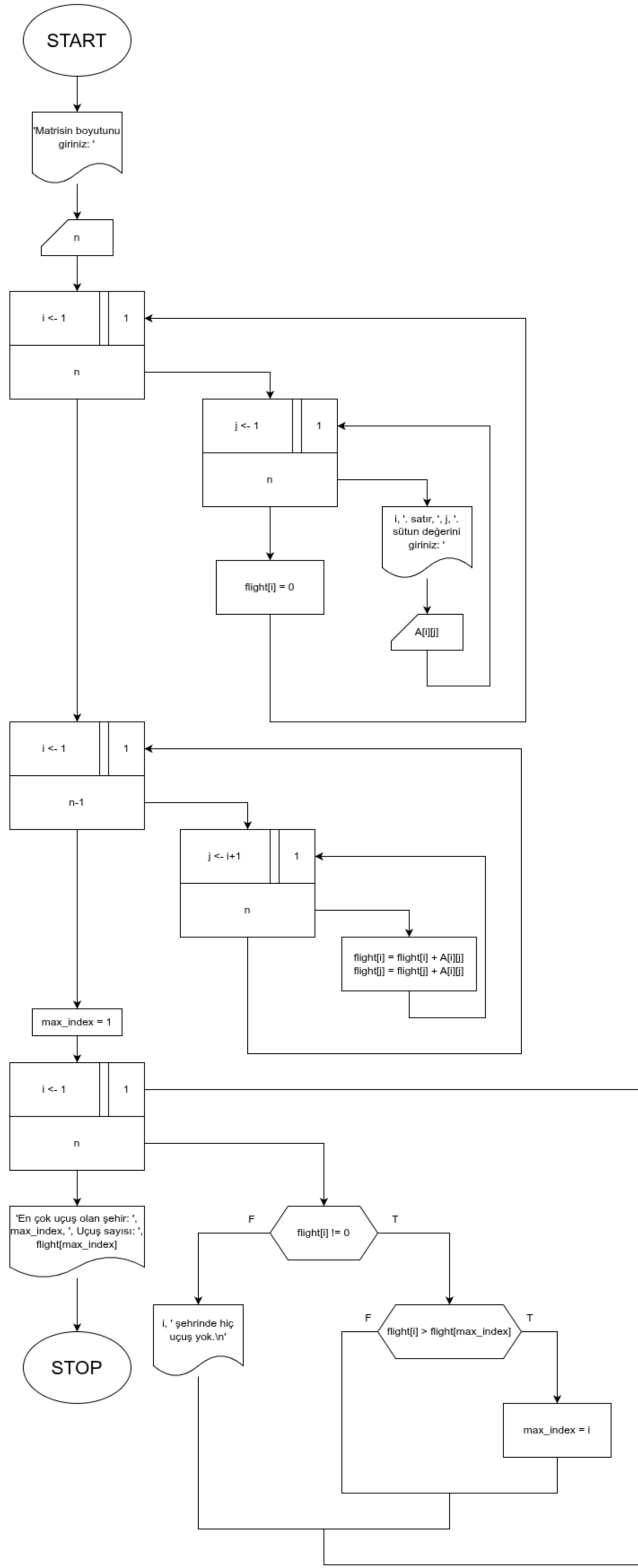
Şehir Sayısı (N): 5

Uçuş Matrisi:

```
0 1 1 1 0
1 0 1 0 0
1 1 0 0 0
1 0 0 0 0
0 0 0 0 0
```

Çıktı:

"5. şehirde uçuş yok."
"En fazla uçuş 1. şehirde, 3 tane."



Soru:

Kullanıcıdan şehir sayısı (N) ve bu şehirler arasındaki uçuş bağlantılarını temsil eden bir matris (komşuluk matrisi) alınacaktır.

- Matrisin (i, j) hücresindeki **1** değeri, i. ve j. şehirler arasında uçuş olduğunu; **0** değeri ise uçuş olmadığını gösterir.
- Uçuşlar çift yönlüdür.

Bu bilgilere göre:

Kullanıcıdan alınan iki farklı şehrin birbirine tek aktarma ile ulaşip ulaşamayacağını bildiren algoritmanın **akış diyagramını (flowchart)** çiziniz.

Örnek Senaryo 1:

Girdi:

Şehir Sayısı (N): 5
Uçuş Matrisi:

```
0 1 1 1 0
1 0 1 0 0
1 1 0 0 0
1 0 0 0 0
0 0 0 0 0
```

Kaynak: 2

Hedef: 3

Çıktı:

"2. şehirden 3. şehre tek aktarma ile varılabilir."

Örnek Senaryo 2:

Girdi:

Şehir Sayısı (N): 5
Uçuş Matrisi:

```
0 1 1 1 0
1 0 1 0 0
1 1 0 0 0
1 0 0 0 0
0 0 0 0 0
```

Kaynak: 1

Hedef: 4

Çıktı:

"1. şehirden 4. şehre tek aktarma ile varılamaz."

