

BLM2512 Veri Yapıları ve Algoritmalar Gr.1-2-3-4-5-6

Ödev – 4

Konu:

Graflar ile Algoritma Tasarımı

Problem:

Verilen yönsüz ve kenar ağırlıksız bir grafta, önceden belirli ancak kullanıcı tarafından bilinmeyen bir düğümde ödül yer almaktadır. Kullanıcı sadece “*X düğümü ödül düğümüne komşu mu?*” sorusunu sorabilmektedir. Ödül düğümünü olası en az soru ile bulabilecek algoritmayı tasarlayınız ve her adımdaki (sorudaki) detayları verecek şekilde çözümünüz için programınızı yazınız. Verilen üç graf için programınızı çalıştırarak çözümünüzü gösteriniz.

Önemli Notlar:

1. Grafi aşağıda verilen formatta dosyalardan okuyabilirsiniz.
2. Grafi, geliştireceğiniz algoritmaya bağlı olarak komşuluk listesinde veya komşuluk matrisinde tutabilirsiniz. Seçim size bırakılmıştır.
3. Okunan bir graf için ödül düğümünün hangisi olduğu kullanıcıdan alınacaktır.
4. *X düğümü ödül düğümüne komşu mu?* sorusunu tasarlayacağınız algoritma soracak ve doğru/yanlış durumuna göre ilerleyerek ödül düğümünü bulacaktır.
5. Algoritmanızın olası en az sayıda soru ile ödül düğümünü bulabilmesi için uyguladığınız stratejinizi ve tasarımınızı videoda açıklamanız beklenmektedir.

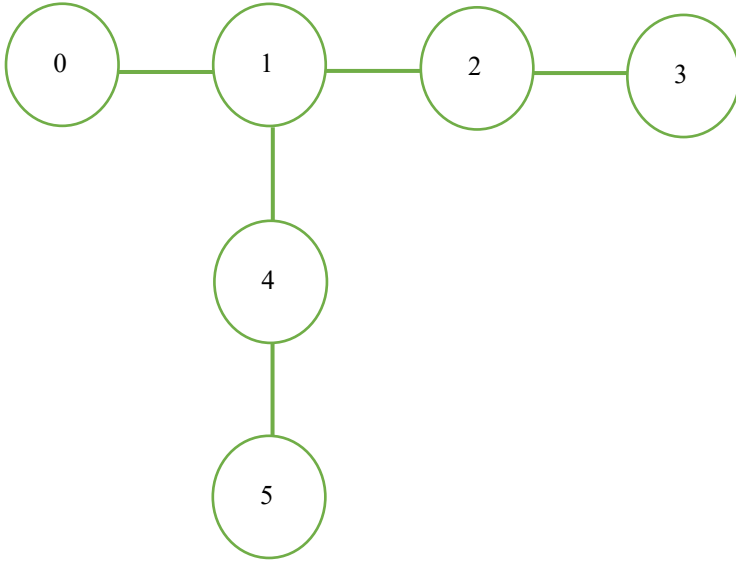
Örnek Çözüm:

Brute-Force yaklaşım ile örnek bir çözüm şu şekildedir.

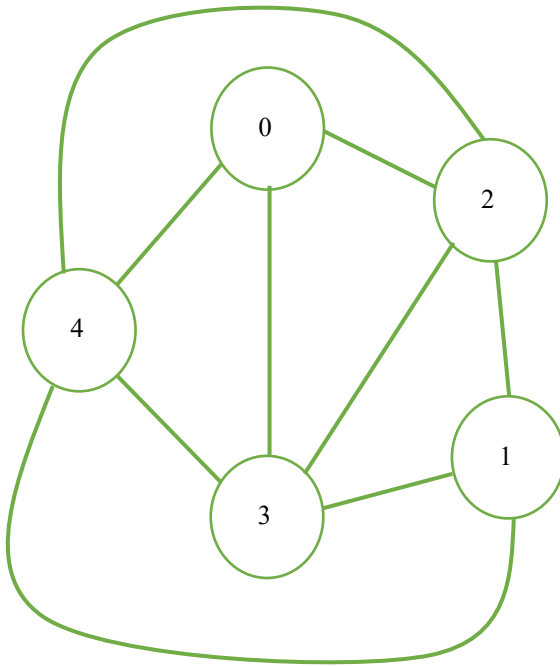
1. `oku(G, "G1KMat.txt")` //Aşağıdaki 1. örnek graf dosyadan okunur.
2. `oku(Vp)` //Kullanıcıdan “5” değeri alınır. ($V_p=5$) (V_p : Vertex_{prize}, ödül düğümü)
3. `odulMu[0..N] = 1` //toplam düğüm sayısı N öncesinde hesaplanmalıdır.
4. `i = 0`
5. `while (i < N)`
 `if !(komsuMu(V[i], Vp))`
 `then`
 `for each v ∈ Adj(V[i]) do odolMu[v] = 0` //düğümün komşuları ödül düğüm olamaz.
 `else`
 `for each v ∈ Adj(V[i]) do if (odulMu[v] != 0) then odolMu[v] = odolMu[v] + 1` //önceden elenmeyen komşuları ödül düğüm olabilir. En fazla doğru cevaba sahip düğüm ödül düğümüdür.
 `end if`
 `i = i + 1`
6. `end while`
7. `yaz("Bulunan ödül düğümü:", indexOfMax(odulMu))` //odulMu dizisindeki en büyük elemanın indisi, ödül düğümüdür.

Kullanılacak Girdi Grafları:

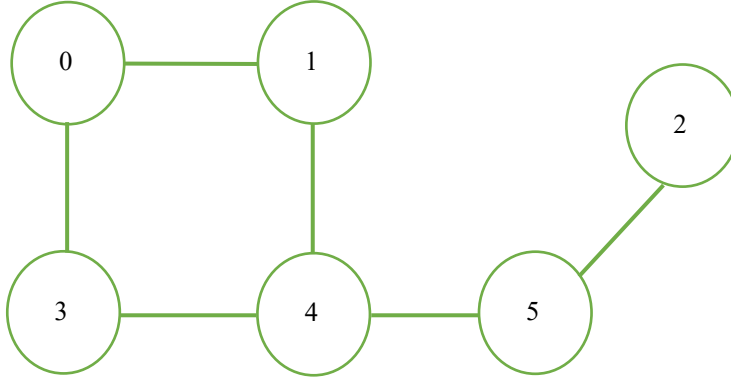
1. Ödül düğümü: 5



2. Ödül düğümü: 3



3. Ödül düğümü: 4



Graf dosya içerikleri:

G1_KMat.txt:

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

G2_KMat.txt:

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

G3_KMat.txt:

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

G1_KList.txt:

```
0 1
1 0 2 4
2 1 3
3 2
4 1 5
5 4
```

G2_KList.txt:

```
0 2 3 4
1 2 3 4
2 0 1 3 4
3 0 1 2 4
4 0 1 2 3
```

G3_KList.txt:

```
0 1 3
1 0 4
2 5
3 0 4
4 1 3 5
5 2 4
```

Ödev Teslimi ile ilgili önemli bilgiler:

Aşağıda verilen bütün bilgileri içeren tek bir doküman hazırlayarak **27.05.2025 saat 23:45'e** kadar online.yildiz.edu.tr adresi üzerinde tanımlı ödev **OgrenciNumarasi.zip** dosyasını yükleyiniz. **HERHANGİ BİR ŞEKİLDE GEÇ ÖDEV TESLİMİ KESİNLİKLE KABUL EDİLMEYECEKTİR.**

Teslim Edilecekler:

1. Anlatılan problemi çözen ana programı ve gerekiyorsa ilgili fonksiyonları içeren programı **C** dilinde yazarak **OgrenciNumarasi.c** dosyasını yükleyiniz.
2. Ekran çıktılarını içeren dosyayı **OgrenciNumarasi.pdf** formatında yükleyiniz.
3. Kısa bir **video** (5-10 dk.) hazırlayınız. Video içeriğinde problemi, geliştirdiğiniz çözümü ve kodunuzu anlatınız. Ardından verilen örnekler ile programınızın çalışmasını gösteriniz. Video linkini raporunuza ekleyiniz. Video linkini public paylaşmayınız, kopyaya sebep olmaktadır.

TESLİM EDİLECEK BELGELER İLE İLGİLİ DETAY BİLGİYİ CLASSROOM'DA PAYLAŞILAN ODEV_KURALLARI.RAR DOSYASINDA BULABİLİRSİNİZ:

Teslim Edilecek Dokümanlar:

- HW#_OgrenciNumarasi.zip (Örn: HW1_25011001.zip)
 - OgrenciNumarasi.pdf (Örn: 25011001.pdf)
 - Uygulama video linki
 - OgrenciNumarasi.c (Örn: 25011001.c)

- E-POSTA ile GÖNDERİLEN CEVAPLAR KESİNLİKLE DEĞERLENDİRİLMEMEYECİTİR.
- DOSYA DIŞINDA SİSTEME HERHANGİ BİR **DRIVE LİNK'i** EKLEMİYİNİZ.
- BAŞKA BİR ÖDEVE VEYA İNTERNET ÜZERİNDE BULUNAN BİR ÇÖZÜME BENZERLİĞİ YÜKSEK OLAN ÖDEVLER KOPYA OLARAK DEĞERLENDİRİLECEKTİR.