Gebze Technical University
Computer Engineering

CSE 321
2017 Fall

HOMEWORK 3

DEVRİM AKINCI
141044052

Course Assistant:Ahmet SOYYİĞİT

QUESTION-1

Devrim Arabaları

1960'lı yıllarda çekilen film dönemin cumhurbaşkanı Cemal Gürsel tarafından istenen Türk yapımı arabanın hikayesini ele almaktadır. Filmin konusu 22 mühendisin 130 günde iki otomobil yapmalarıdır. Filmde Gündüz isimli mühendisin seçmiş olduğu 22 tane mühendisin 130 günde imkansız olarak bakılan araba yapımı için Eskişehir'deki fabrikada birleşmesi ile başlanmıştır ve arabanın Cumhuriyet Bayramı'na yetiştirilmesi gerekmektedir. Devletteki bazı bakanların bu konuda ekonomik sebepleri göz önüne alarak bu işe karşı çıkmalarına rağmen mühendislerin canla başla çalışarak, gazetelerde çıkan karalama haberlere karşın büyük mücadeleyle araba yapımlarını sürdürdükleri gözlenmiştir. Cemal Paşa'nın gelmesinin 3 gün önceden haber verilmesi sebebiyle alelacele motoru birleştirmeye çalışmışlardır ve gelmesine 1 gün kala sorun çıktığını gözlemlemişlerdir. Bunun üzerine başmühendis Gürbüz Bey'in konuşma yapması ile tekrar canlanan mühendisler Cemal Paşa'ın önüne çalışan bir motor çıkarmayı başarmışlardır. Daha sonrasında büyük zorluklara rağmen araba yapımını tamamlayan mühendisler arabayı trenle götüreceklerdir fakat trende arabanın üstüne ateş parçası sıçraması ihtimaliyle arabanın benzinini azaltmak zorunda kalmışlardır. Bu durum Cemal Paşa'nın köşkten alınmasıyla devam eden süreçte büyük soruna yol açarak Cemal Paşa'nın köşkten alınıp arabaya bindirildikten 100 metre sonra yolda kalmasına sebep olmuştur. Bunu haber yapan gazetecilerden dolayı halk arabayı sahiplenememiş, arabaların yapımına daha fazla ekonomik destekte bulunmamışlardır. Filmin başında müzeyi ziyaret eden kişi filmin içerisinde rol alan mühendislerden biri olan Cemal'dir. Bu arabaların yapımının durdurulması onu da çok üzmüştür. Arabalardan biri hurdaya çıkmıştır. Sadece bir tanesi Eskişehr'deki müzede durmaktadır.

QUESTION-2

İlk olarak asistan sayısına göre permütasyon hesabı yaptım. Eğer asistan sayısı ders sayısından büyükse permütasyon fonksiyonun output olarak verdiği listenin içerisindeki elemanı, artan asistan sayısına göre değiştirdim. Örneğin 4 asistan 3 tane ders olsun. Derslerin kodları (0 , 1 , 2) olacak. Asistana göre permütasyon alırsak [0,1,2,3],[0,1,3,2],.....,[3,2,1,0] diye 24 tane sonuç gelecek. Burada 3 değerini -1 yaptım çünkü artan asistan sayısı bir tanedir. Daha sonra permütasyon fonksiyonun output olarak verdiği listeyi dolaşarak her bir olasılığın cost'unu buldum ve bunları bir listeye attım. O listenin minimum değerini buldum ve o listede minimum değer hangi index ise o indexi buldum. Index fonksiyonu ile permütasyon listesinde bulduğum indexi vererek minimum costun olduğu durumu bularak returnList ' e atadım. returnList ile minimum 'u return ettim.

Complexity Analysis

Best Case : Eğer bir asistan ve bir ders varsa best case meydana gelir.

$B(n) \in \theta(1)$

Worst Case : Eğer asistan sayısı ders sayısından daha büyükse veya eşitse worst case meydana gelir.

Permütasyon fonksiyonun karmaşıklığı θ (n!) 'dir. Permütasyon fonksiyonun output olarak verdiği listenin içerisindeki listeler n elemanlıdır. Bu listeleri iterative olarak gezindiğinde Worst Case -> $W(n) \in \theta(n*n!)$ olur.

QUESTION-3

Grafı gezmek için breadth first search (BFS) algoritmasını kullandım. Bu algoritmayı kullanmadan önce bana lazım olan veri yapısı Queue'yu implement ettim. BFS'yi implement ederken verilen vertex'den başladım. İlk vertex'i queue'a koydum ve koyduğum vertex'i visitList'e ekledim. Queue boş olmadığı müddetçe queue'dan çıkardığım elemanın tüm komşularına baktım ve bu komşular visitList'de değilse queue'a ve visitList'e ekledim. Eklerken de road değişkenimi bir arttırdım. Bu işlemi queue boş olana kadar devam ettirdim. Fonksiyonun sonunda visitList ile road return ettim. Grafın disconnected vertexleri bulmak için graftaki bütün vertexleri BFS ile gezdim. BFS'nin return ettiği visitList'i alıp sort ederek yeni bir liste elde ettim ve bunu tuple çevirdim. Daha sonra bu tuple'ı set'e ekledim. Set'e eklediğim tuple'nin ilk elemanlarını bir listeye attım ve bu listeyi return ettim. findMinimumCostToLabifyGTU fonksiyonunda eğer lab inşa etme maliyeti yol yapma maliyetinden düşükse bütün vertexlere lab inşa ettim. Aksi durum var ise, ilk önce findDisconnectedVertex bu fonksiyonu çağırarak disconnected vertexleri bulup o vertexleri sırası ile BFS ile gezdim. BFS'nin return ettiği road'u listeye attım. Son olarak roadList'de bulunan değerleri sırası ile yol yapma maliyeti ile çarparak topladım. Elde edilen sonucu disconnectedList'de bulunan değerleri sırası ile lab inşa etme maliyeti ile çarparak topladım. Elde edilen sonucu return ettim.

Complexity Analysis

Best Case : Eğer grafta tek bir vertex varsa best case meydanaa gelir.

$B(n) \in \theta(1)$

Worst Case : Eğer bilgisayar laboratuvarı inşa etme maliyeti yol yapma maliyetinden büyükse o zaman worst case meydana gelir.

Disconnected vertexleri bulmak için grafın tüm vertexlerini iterative ediyorum. Iterative ederken BFS yi çağırıyorum. BFS nin karmaşıklığı O(|V| + |E|) dir. Grafın vertex sayısı V olduğuna göre, Worst case -> $\theta(V^2 + VE)$ V^2 daha büyük olduğu için $W(n) \in \theta(V^2)$ dir.

QUESTION-4

Insertion Sort

Step-1

12	34	54	2	3

12 < 34 olduğu için swap yoktur. Bir sonraki ikili kontrol edilir.

Step-2

12	34	54	2	3

34 < 54 olduğu için swap yoktur. Bir sonraki ikiliye bakılır.

Step-3

12	34	54	2	3

54 > 2 olduğu için swap edilir.

Step-4

12	34	2	54	3

34 > 2 olduğu için swap edilir.

Step-5

12	2	34	54	3

12 > 2 olduğu için swap edilir.

Step-6

2	12	34	54	3

Iteration kaldığı yerden devam eder. 54 > 3 olduğu için swap edilir.

Step-7

2	12	34	3	54

34 > 3 olduğu için swap edilir.

Step-8

2	12	3	34	54

12 > 3 olduğu için swap edilir.

Step-9

2	3	12	34	54

Sıralı bir dizi

Shell Sort

Birden fazla Insertion sort kullanılır. Bir gap sequence belirlenir. Buna karar vermek için Hibbard's Sequence kullanılabilir.

Hibbard's Sequence: $2^k-1 <= n$ k = Interval value n = Listenin eleman sayısı

Sıralanacak liste 5 eleman olduğuna göre k = 3 ' tür.

Step-1

12	34	54	2	3

k=3, 12 > 2 ve 34 > 3 olduğu için swap edilir ve k'nın değeri 2 azaltılır.

Step-2

2	3	54	12	34

k=1 olduğu için normal insertion sort yapılır.

Step-3

2	3	54	12	34

54 > 12 olduğu için swap edilir.

Step-4

2	3	12	54	34

54 > 34 olduğu için swap edilir.

Step-5

2	3	12	34	54

Sıralı bir dizi

Insertıon Sort vs Shell Sort

Insertion sort time complexity:

-Best Case: $\theta(n)$

-Worst Case : $\theta(n^2)$

-Average Case: $\theta(n^2)$

Shell sort time complexity:

-Verilen gap sequence göre time complexity de değşir.

-Hibbard's sequence göre: $O(n^{1.5})$

-Shell sequence göre: $O(n^2)$