

Soru 1:

Toplu iğneyi dahi dışarıdan ihtal eden bir ülkede, cumhuriyet bayramına yetiştirmek üzere Cemal Paşa milli araba yapılması emrini veriyor ve neredeyse imkansız olan bu görevi gerçekleştirebilecek kapasiteye sahip bir ekip arayışına giriliyor ve Gündüz Serter’de karar kılınıyor. Gündüz Serter bu görevi memnuniyetle kabul ediyor ve güvendiği mühendislerden oluşan bir ekip kuruyor. Eskişehirde çalışmaya başlıyorlar, gerekli teknik aletleri ise yok, sadece basit bir vinç ve küçük el aletlerine sahipler çünkü muhalifler ve basın bu projeye karşıdır, bu ekibe oldukça cüzi bir miktar vererek, fazla sına gerek yok bu proje gereksiz düşüncesindelerdir. Bütün zorluklara rağmen proje maketlerle başlayıp hızla ilerliyordu, karşılaşılan zorluklara milli ve pratik çözümler üretiliyordu. Bu ilerlemer Sami ve yandaşlarını ise oldukça endişelendiriyordu, çünkü onlar bunun bir hayal olduğunu, eğer proje başarılı olursa Cemal Paşa’nın bir hevesle fabrika kurulmasını isteyeceğini ve bu fabrikanında daha sonradan unutulup kapatılacağını, bu işler için tüm masrafların boşuna olduğunu, ülkenin bu durumu kaldıracak güçte olmadığına inanıyorlardı. Latif Bey ise projedeki mühendislerden birisi olmasına rağmen (ki yakın arkadaşı Gündüz projeye katılmasını istediği için katılıyor projeye) proje hakkında pek ümitli değildi, arabayı yapsalar bile hiçbir şey olmayacağını düşünüyordu çünkü Atatürk’ün kurduğu tayyare fabrikasının üstün başarılarına rağmen (ki bu başarılar Hollanda’ya uçak satmaya kadar gitmiş) kapatıldığını ve traktör fabrikasına döndürüldüğünü biliyordu çünkü o, zamanında tayyare fabrikasında yetişmişti ve bu durumun hiçbir zaman değişmeyeceğini, Türkiye’de hiçbir başarının cezasız kalmayacağını düşünüyordu. 1961 yılında kendi yerli ve milli imkanlarıyla bütün zorluklara rağmen araba üretiliyor ve 2 Devrim arabası trene bindirilip Ankara’ya gönderiliyor, trende güvenlik gerekçesiyle arabaların benzinleri boşaltılıyor. Tören zamanı geliyor ve Cemal Paşa birinci arabayı denediğinde benzin koyulamadığından araba 100 metre anca gidiyor ve duruyor, bunu gören basın hemen “Başarısızlık” olarak projenin anti reklamını yapıyor, Cemal Paşa ikinci arabaya geçiyor ve hiçbir sorun yaşanmıyor, araba gerçekleştirmesi gereken tüm fonksiyonlarını başarıyla gerçekleştiriyor fakat dönemin basınına bu projeye karşı baştan beri olan karalama kampanyası ile proje karalanıyor ve çok hüznü bir şekilde projeyi başarısızlıkla sonlandırılıyor.

2.Soru:

Öyle bir kombinasyon bulmalıyız ki bu kombinasyona göre asistanlar kendilerine atanan derslere en az süreyi harcamalı. Bu düşünceye göre, r tane asistan, n tane ders varsa, r elemanlı bir liste oluşturduğumuzda, bunun n 'li kombinasyonları, derslere atanacak farklı asistanların tüm durumlarını oluşturur. Bu duruma göre, eğer ders sayısı asistan sayısından daha büyükse "None" return ediliyor. Diğer durumlarda, asistan sayısı kadar bir liste oluşturuluyor, örneğin asistan sayısı 5 ise [1,2,3,4,5] olarak bir liste oluşturuluyor ve bu listenin ders sayısı kadar kombinasyonları oluşturuluyor, bu oluşturma şu şekilde oluyor. Asistan listesinin tüm alt kümeleri bulunuyor daha sonra bu alt kümelerden course sayısı kadar olanlar seçiliyor. Yani $\binom{ra\ list}{n}$ kombinasyonları bulunuyor. Daha sonra bulunan bu kombinasyonların her biri için tek tek cost hesaplanıyor ve en düşük cost ile bu cost'un permütasyonu bulunuyor. Daha sonra asistan sayısı 0 dan başlaması gerektiği, bizim ise 1 den başlattığımız için tüm elemanlardan 1 çıkartıyoruz. Ardından, bu permütasyon ile asistanların vereceği derslerin numaraları elde ediliyor. (Örneğin permütasyon [0, 1, 3, 2, 5] ise ve 6 asistan varsa, 0. Ders, 0. Asistana, 1. Ders 1. Asistana, 2. Ders 3. Asistana, 3. Ders 2. Asistana, 4. Ders 5. Asistana ait demektir. Bu durumda 4. Asistan ders ders vermemektedir ve return edilecek liste şu hale gelmektedir: [0, 1, 3, 2, -1, 4])

Complexity hesabı,

Faktöriyel fonksiyonunun complexitisi $n!$ 'dir. Bunu "Create_perms_for_n_elems_subset" fonksiyonunda çağırıyoruz, bu fonksiyon ise $\binom{r}{n}.r!$ complexityye sahip, findOptimalAssistantship ise

"Create_perms_for_n_elems_subset" fonksiyonunu çağırarak döndü değerine göre bir döngü üretiyor, içerideki döngü n 'lik işlem yapıyor, döngüyle beraber ise $n.\binom{r}{n}.r!$ complexity olarak söyleyebiliriz main fonksiyonu. (düşük katsayılar ihmal edildi.)

3. soru

Unconnected componentlar için şöyle bir çözüm getirdim: BFS bana gezmeye başladığım node'den ulaştığım tüm nodeler için parentlarını verecek, ulaşamadıklarına ise -1 verecek. (başlangıç nodesine 0 verdirttim ulaşamadıklarıyla farklı olmaları için). Yani BFS kendisine verilen start vertexinden gezemediği tüm vertexler için -1 değeri vericek. Örneğin 1. Vertexten BFS başlattığımda [0,1,1,2,-1,-1] gibi bir array verirse bunun anlamı 5. Ve 6. Vertexlere, 1. Vertexten erişemediği yani 5. Ve 6. Vertexlerin unconnected component olduğudur.

Bunu garantiledikten sonra recursion ile cost'u hesapladım. Bu recursiona göre, bir starting point ile BFS çağırılıyor ve parent arrayi alınıyor, alınan bu parent arrayinde -1 olmayan indexler için visited arrayindeki indexler True yapılıyor ve o subgraphta kaç adet yol olduğu hesaplanıyor ardından "(yol sayısı * yol maliyeti + lab maliyeti + recursion fonksiyonu)" return ediliyor. Recursion fonksiyonu return ediliyor, çünkü bulunan subgraphlarla işlemi o recursion fonksiyon yapacak. Recursion fonksiyon tüm nodeler visited edildikten sonra ise duruyor.

Bu şekilde 1 lab ve ona ulaşan yolların tamir costu hesaplanıyor, ardından bu hesap, sadece tüm lab lar tamir edilseydi ne kadar cost olurdu? Sorusunun cevabıyla karşılaştırılarak, en düşük cost hangisiyse o return ediliyor.

Complexity,

BFS adjanceny list kullandığımızdan $O(V+E)$ complexitiye sahip, bu fonksiyonu recursion fonksiyonumda her ayrı bir subgraph için çağırdığımdan en kötü durum subgraphların fazla olduğu durumdur. Maksimumda $V.O(V+E)$ ye kadar gider worst case.

4.Soru

Insertion sort:

12, 34 , 54, 2, 3	12<34 OK
12, 34, 54 , 2, 3	34<54 OK
12, 34, 54, 2 , 3	54>2, 34>2, 12>2 SWAP
2, 12, 34, 54, 3	54>3, 34>3, 12>3 SWAP
2, 3, 12, 34, 54	SORTED

Shell sort:

$$\text{GAP} = n/2 = 5/2 = 2$$

12 , 34, 54 , 2, 3	12<54 OK
12, 34 , 54, 2 , 3	34>2 SWAP
12 , 2, 54 , 34, 3	54>3, 12>3 SWAP
3 , 2 , 12, 34, 54	GAP = $n/4 = 5/4 = 1$ AND 3>2 SWAP
2, 3 , 12 , 34, 54	3<12 OK
3, 2, 12 , 34 , 54	12<34 OK
3, 2, 12, 34 , 54	34<54 OK

GAP = $n/8 = 5/8 = 0$ So array is SORTED.

Shell sort, birbirinden uzak indexlerdeki elemanları kıyaslar bunun anlamı, elemanların bulunması gerektiği yerlere daha yakın olmasını sağlamasıdır. Bu uzak indexleri kıyaslama ölçütü (GAP) git gide azalarak 1'e kadar iner, Shell sortun yapmak istediği GAP 1'e inmeden tüm değişiklikleri yapmış olmak (ya da çok az değişiklik yapmak bu GAP 1 iken) ve arrayi sıralamaktır. Böylece her seferinde her eleman her elemanla sürekli yer değişiminde bulunmaz.

Insertion sortta ise, yeni eklenen eleman eğer en son elemandan küçükse, yeni elemanın yerleştirileceği yer bulunana kadar, listedeki elemanlar ile tek tek karşılaştırılıyor, buda oldukça masrafa yol açıyor.

Best case durumunda (arrayin sıralı olması) Shell sort $n \log n$ iken, insertion sort n dir. Yani insertion sort daha iyidir, average case de ve worst case de Shell sort $n(\log n)^2$ iken, insertion sort n^2 dir, bu durumlarda Shell sortun farkı ortaya çıkar, insertion sort'a göre çok daha hızlı sonuç verir.

Insertion sort online sorting algoritması iken, Shell sort online değildir. Buda Shell sortun büyük bir eksisidir.