CSE 321 – Introduction to Algorithm Design
==Homework 3==

QUESTİON 1:

Film; 1960 yıllarda geçen , konusu bir avuç mühendisin verilen zor görevi kısa zamanda bitirilmesi gerekir ve başarırlar.

Dönemin cumhurbaşkanı Cemal Gürsel bu milli ve önemli görevi Eskişehirde'ki TCDD verilir ve orada 20 kadar mühendis işe başlarlar . Verilen bu görevin en zor kısmı ise vaktinin dar olmasıdır . Çünkü arabanın yapılıp test edilmesi için toplamda 130 günleri vardır ki bu günde 29 Ekim'e denk gelmektedir . İlerleyen zamanlarda birçok aksaklıkla uğraşırlar hatta mühendisler bir araba yapmayı planlarken 2. bir emir ile başka araba siparişide verilir . Son güne kadar çalışan mühendisler arabaları Ankara'ya götürmek için trene yüklerler ve bi patlama olmasın diye depolar boşaltılır . Mühendisler sadece bir arabanın deposunu doldurabilmişlerdir . Dolu olan araç önden gitmiştir . Cumhurbaşkanı ise arkadaki araca binmistir ama araç 100 metre gittikten sonra deposu boş oldugu ıcın durmustur . Bu durumda ülkede büyük yankı uyandırmıstır .

QUESTİON 2:

Best Case: Eger ki R degeri N degerinden kücük ise bos liste return eder bu da constant time zamandadır .

Avarage Case: Subsetleri uretmekten daha pahalı bir process olmadıgı ıcın wortscase ıle aynı durum .(R!*N)

Worst Case: Butun subsetleri uretirken R!*N karmasıklıgı vardır asıl ıs yuku burda oldugu ıcın burayı ıllakı yapıcak program.

Explaining the algorithm:Öncelikle r ve n karsılastırması var ve bu constant time(best case) zamanda olur. R < N cıkarsa program dırek sonlanır ve bos kume ve minTİme olarak da 1000 dondurur(burdaki 1000 degeri minTime in ilk degeri oldgu ıcın daha sonra degisiyor.). Daha sonra olasılıklar kümesi icin listeye bos liste ekler inputTable kadar (O(n)1). Daha sonra olasılıkları üreten fonksiyonun icinde ana döngü icinde recursion var burdada Asistan yani R degeri azalarak ıslem gorur ve donguyu devam ettirende N degeridir . Burdanda karmasıklık R!*N .Programın devamında R lerin indislerini hesaplamak için fonksiyon cagırılır . Burdada ic ice 2 tane for dongusu vardır buda O(n^2) karmasıklık eder . Daha sonra
tekil 2 tane for dongusu var.Donguleın ıcınde karsılastırma var ve donguler n karmasıklıgında ve içerdeki karsılastırmalarda constant time zamanda olur.Sonuc olarak programın en karmasık kısmı subsetleı uretme durumudur.

QUESTİON 3 :

Worst Case: Graphın eleman sayısı kadar dönen for icinde dfs cagırılıyor .Dfs fonskyonunun ıcındede while dongusu var bu da graphın elemanın komsu sayısı kadar donuyor. Yani graphın eleman sayısı ile elemanlarının icinden en cok komsuya sahıp olanın carpımı kadar bır karmasıklıgı vardır. Yani O(n*max(n)) karmasıklıgı olabılır .

Explaning the algorithm : Algoritma en once graphın elemanlarının listeye atar (setlerden kurtulmak icin) burda tekil for dongusu vardır(O(n)).Daha sonra for dongusu ıcınde dfs ve isHave fonksiyonu cagırlır . Dfs fonksşyonu ıcersınde while dongusu vardır . Bu dongu en fazla graphın elemanlarından maximum uzunluga sahıp olan kadar donebılır . Pythonun sort() fonksiyonuda kullanılmıstır(hangi sortu kullanıyor bılmıyorum o yuzden onemsız kabul edıyorum bu sortu). Daha sonra isHave fonksıyonu cagırılır bunun ıcersındede tekıl for dongusu vardır . En sonundada lab ve road malıyetleri hesaplanır burasıda tek for ıcınde olur .

QUESTİON 4:

Insertion Sort

1. | 12 34 54 2 3
2. 12 | 34 54 2 3
3. 12 34 | 54 2 3
4. 12 34 54 | 2 3
5. 2 12 34 54 | 3
6. 2 3 12 34 54 |

İnsertion sort , performans olarak yavaş ama programlama olarak basit bir yapısı vardır .
İlk indisten başlayarak sağ tarafa doğru her elemanı sol tarafta sıralı yerine sokar . Bu şekilde en sağ tarafa gelindiğinde liste sıralanmış olur .
Performansı O(N^2) dir . Bunun sebebi dizidideki eleman sayısı kadar Geçiş yapması gerekir ve her geçişte worst case durumunda seçili eleman yine n kadar kaydırılır .

Shell Sort

1. 12 (34) 54 (2) 3
2. 12 2 (54) 34 (3)
3. (12) 2 (3) 34 54
4. (3) (2) 12 34 54
5. 2 3 12 34 54

Shell sort , atlama miktarı belirlenir . En basit yöntem listedeki eleman sayısının yarısı kabul edilir . Sırasıyla her sayı kendinden 3 sanraki sayı ile karşılaştırılır ve swap yapılır . Daha sonra atlama miktarı mevcut atlama miktarının yarısı konumuna getirilir .Performansı O(n^3/2) dir . Shell sort cok uzaktakı elemanları degısmesını saglar insertion sorta gore daha avantajlıdır .