

# CSE 321 – Introduction to Algorithm Design

HOMEWORK 3

RIDVAN DEMİRCİ

141044070

## DEVİRİM ARABALARI

Devrim Arabaları 1960 lı yıllarda yönetime darbe ile gelen Cemal Gürsel'in ülke içi ve dışındaki insanların otomobil yapar diye söyledikleri söze yanıt olarak , "Türk mühendisleri otomobil yapacaktır" demesi üzerine ulaştırma bakanlığına verdiği emir ile mühendis takımı kurulur ve Eskişehir'de çalışmaya başlarlar. Tüm olumsuz şartlara rağmen araç için çalışmalar başlar ,bu sırada ülke içinde ki bazı bakanlar bile "ülkenin sokağa atacak parası yok" diye proje destek vermemiştir ve her türlü olumsuz şartları oluşturmuşlardır. Her şeye rağmen iki araç 29 Ekim günü tamamlanmış ve meclis önüne getirilmiştir, ancak trende tedbir nedeniyle benzin depoları boşaltılmıştır yani meclis önünde iken siyah olan aracın deposu boştur. Cumhurbaşkanı gelir arabaya biner ve 100 metre felan ilerledikten sonra araba benzin bittiği için durur ve ilerlemez sonra Cemal paşa benzin den dolayı gitmediğini öğrenir araçtan iner ve olanlar için

**"Garb kafasıyla otomobil yaptık ama şark kafasıyla benzin koymayı unuttuk" ,der.**

Ve haberlere devrim yolda kaldı diye haberler düşer daha sonra proje iptal edilir.

## findOptimalAssistantship

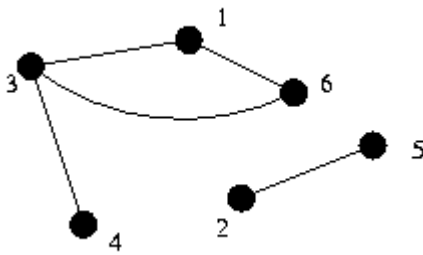
---

- Problemin çözümü için öncelikle permutation fonksyonu nu heap permutation algoritmasını kullandım, bunun için pseudocode'dan yardım aldım([https://www.wikizero.com/en/Heap's\\_algorithm](https://www.wikizero.com/en/Heap's_algorithm))
- Analiz hesabı için ise asistan sayısı az ise -1 return eder hesabı katılır ise bestcase  $\theta(1)$  gelir
- Ancak hesaba katılmaz ise ;  
Methodun içindeki permutation hesabı  $r!$  Kadar sürer daha sonra üzerinde gezinme işlemide  $r!$  Kadar zaman alır ancak bu işlem sequence olduğu için toplamda  $r!$  Zaman sürer asimtotik notasyon olarak ise  $\theta(r!)$  kadar zaman alır

$$T(N) = \theta(r!)$$

## findMinimumCostToLabifyGTU

- Eğer yol labdan daha masraflı ise her node için lab yapılır yani vertex sayısı kadar lab yapılır bestcase .
- Eğer yol daha az masraflı ise, Bu problemde öncelikle gelen graphın connected veya not



connected olmasına bakılır  
connected değil ise başlangıç node  
ları listede return edilir örnek  
olarak ;

Yandaki graph da `checkConnected`

Methodunun return değeri [1,2] olacaktır not connected vertex başlangıçları

Daha sonra **findMinimumCostToLabifyGTU** methodunda ise gelen vertex değerlerinden başlayıp tüm vertexleri dolaşır ve o vertex için bütün node sayısını bulur. o graph için 1 tane lab vertex sayısı -1 tane de yol yapılır.

- Eğer yol labdan masraflı ise yani  $x < y$  ise her departmana lab yapılır hiç yol yapılmaz algoritma toplam vertex \* lab masrafı olur constant time
- Eğer  $x > y$  ise breadth first search algoritması ile tüm vertexleri gezer bu işlem de vertex ve onun edgeleri ile alakalıdır yani breathfirst search complexitiy ile aynıdır.  **$\Theta(E+V)$**

$$\begin{cases} \Theta(1) & , & x < y \\ \Theta(E + V) & , & x \geq y \end{cases}$$

## Shell Sort and Insertion Sort

**Data = { 12, 34, 54, 2, 3 }**

### 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. 12,34,**2**,54,3
6. 12,**2**,34,54,3
7. **2**,12,34,54,3
8. 2,12,34,54,**3**
9. 2,12,34,**3**,54
10. 2,12,**3**,34,54
11. 2,**3**,12,34,54

Shell Sort: başlangıç olarak sıralama boşluğu (**gap**)

Size/2 secilir.

Gap =  $5/2 = 2.5 = 2$ ,

1. **12**,34,**54**,2,3
2. **12**,34,**3**,2,**54**
3. **3**,34,**12**,2,**54**
4. 3,**34**,12,**2**,54
5. 3,**2**,12,**34**,54

Array gap 2 ye göre sıralandı şimdi gap tekrar 2 ye bolunur.

Gap = gap / 2 ,**gap =1**

6. **2**,**3**,12,34,54

Array sıralanmış olur ve daha swap işlemi olmaz

Yukarıda kutularda da görüldüğü gibi Shell sort uzakdaki değerleri önceden sıraladığı için **swap** işlemi gap 1 olduğu için daha az olur bu yüzden time complexity insertion sort'a göre daha

efficient'dir.yukarıdaki örnekte Shell sort 6 iterasyonda tamamam insertion sort ise 11 iterasyonda sıralanır.

Shell sort gab aralığına göre değiştiği için analizi hesaplanamaz ama en efficient durumu gap = 2.7 ile bölünürse average time karmaşıklığı  $\theta(n^{5/6})$  olur.

Shell sortda swap işlemi insertion sorta göre daha az gerçekleşir.