

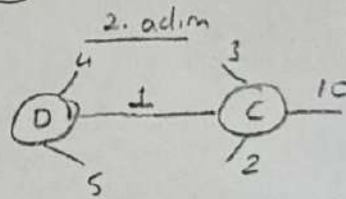
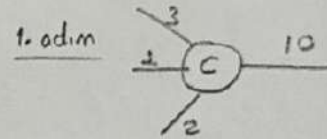
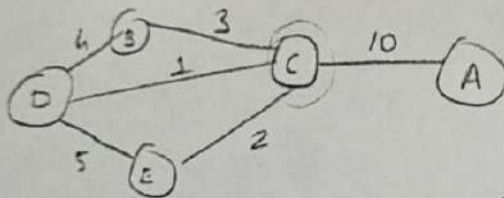
①-P: Polinom zamanda 45 sorular

NP: Polinom doğrulanabilir

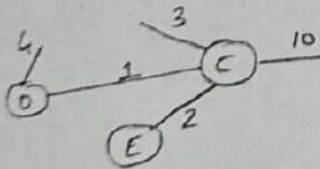
NP-complete:

P polinom zamanlı algoritmaları, NP polinom zamanında doğrulanabilir, NP-complete burada da zor problemleri, NP-hard da çok daha zor problemleri ifade eder

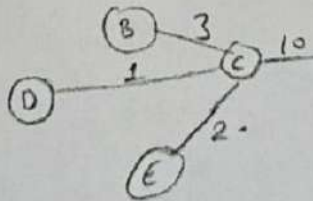
③- Yanda verilen graf için MST yi Prim ile bul. Başlangıç C her adımda dist array değişimini belirt



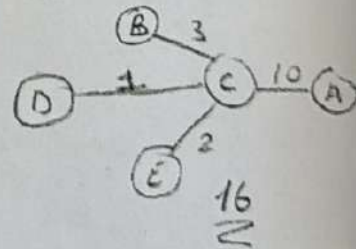
3. adım



4. adım



5. adım MST



Kapasitesi sığan bir sırt çantası dinamik programlama

a) Sırt çantası problemi için yineli bir bağlantı

b) dinamik programlama tablosu

c) Zaman ve yer karmaşıklığı

Nesne	Ağırlık	Kazanç
1	2	12
2	1	10
3	3	20
4	2	15

b) Kütünden kapasite

	0	1	2	3	4	5
1	0	0	12	12	12	12
2	0	10	12	22	22	22
3	0	10	12	22	30	30
4	0	10	15	25	30	35

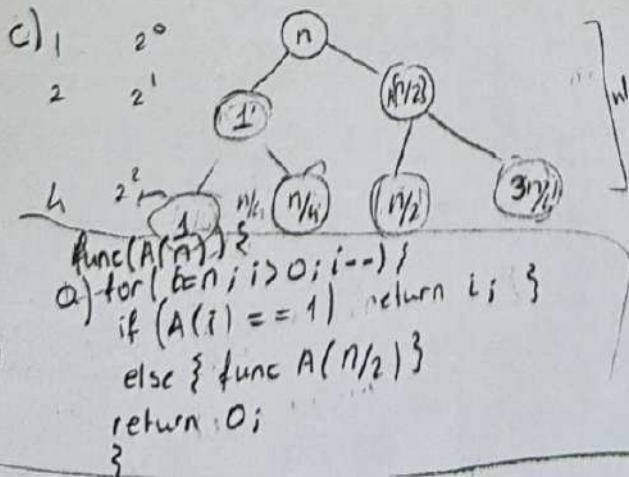
c) $T(n, w)$ ^{item} ^{ağırlık}

$S(n, w)$

d) Pozitif tam sayılardan oluşan önceden sıralanmış $A[1, \dots, n]$ dizisi

verildiklerinde eğer dizide $A[i] = 1$ eşitliği var ise index i 'yi çıktı veren değil de -1 döndüren bir algoritma

a) Perçak ve kaset b) zaman karmaşıklığı alt yarımlara bölünür, c) Zaman



adım b)

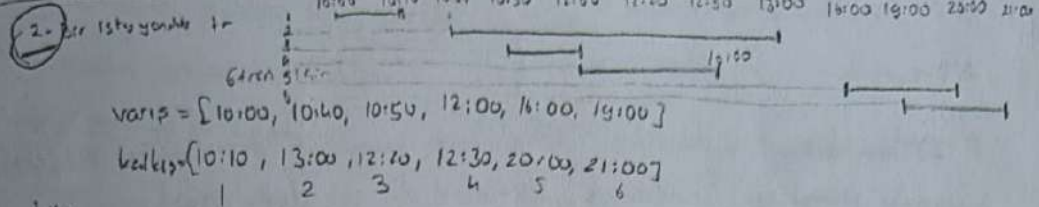
$T = O(n \log n)$

$S = O(1 + 2^1 + 2^2 + 2^3 + \dots + 2^n)$

Formülü kullan

karşılaştırmalı
yöntem
diğer
yöntem

1- P, NP, NP-complete ve hard karmazıklık sınıflarının farkı kısaca açıklayınız



0) Kaba kuvvet tescillenmesi olabileceği tüm alt kümelerin karşılaştırılması gerekir
 bu da $T = O(2^n)$ ile döngü veriyor
 yer karmazıklığı n elemanlı ile doğru olduğundan $S = O(2n) = O(n)$

b) Aşağıdaki algoritma

```

sayac = 1
istasyon = 1
son = liste[0]
for i = 1 to n
    x = liste[i]
    if feasible(x, liste[0]) then
        sayac++
        istasyon += sayac
return istasyon
    
```

bitiş zamanına göre algoritma (kalkış)

Sıra

liste → 1 - 3 - 4 - 2 - 5 - 6

burada yeni elemanın varış zamanı ile önceki elemanın kalkışı ile karşılaştırarak kontrol etmeliyiz. Diğer elemanlar için istasyon etki

c) $T = O(n)$
 $S = O(n)$

d) Optimum çözümün garanti eder mi? Garanti etmez

Aşağıda 2 istasyon yeter ama greedy yöntemi en iyiyi bulup diğerleri sayısınca istasyon ekler yani 4 istasyon yapar.

