



**Kurs Adı:** Algoritma Analizi

**Öğrenci Adı ve Soyadı:** Arda Kaşıkçı

**Ödev Konusu:** Dinamik Programlama

## Yöntem:

Öncelikle verdiğiniz dosyadan çektiğim satırları bir struct dizisine alıyorum. Sonrasında bu diziyi bitiş zamanlarına göre sıralanacak şekilde merge sort algoritmasına sokup, sıralıyorum. Sıralama işleminde satır numaralarını kaybedeceğimden dosyadan çekerken struct'ın içine reklamNo kaydettim.

Sonrasında **max\_profit** isimli fonksiyonla bu sıralanmış reklamları tek tek dinamik programlama gereği bir matrise almaya başlıyorum. Matrisin 0. Satırını 0'lıyorum ki sonraki işlemlerde profit[0]'ı kullanabileyim. Sonrasında tek tek satırları eklerken şuna dikkat ediyorum:

- 1) Elimdeki reklamda önce biten ve elimdeki reklamla çakışmayan son reklamı buluyorum.
- 2) Son çakışmayan reklamın profitiyle, yeni reklamın value değeri toplanır.
- 3) Ardından bir önceki satırdaki profit değeriyle yeni hesapladığımız değer karşılaştırılır.
- 4) Yeni hesaplanan değer eski profitten büyükse yeni değer profit kısmına yazılır. Büyük değilse eski profit değeri yazılır ve bu şekilde devam eder.
- 5) Ayrıca Re-Construction işlemi için her satırdaki profit değerine yönelik bir label dizisi tuttum. Label[i]=0 ise o satır bir önceki adımdaki profit hesabını kullanmıştır. Label[i]=1 ise o satır hesaplanan profit değerine dahildir.

Ardından matris kurulduktan sonra matrisin sonundaki değer bizim max profitimizi verir.

Bu değer hangi reklamlardan geliyor diye bulabilmek için tuttuğumuz label dizisiyle beraber adım adım geriye gidilerek, mak profit sağlayan reklamlar bulunur.

### **Karmaşıklık**

Zaman Karmaşıklığı:  $O(n \cdot \log n) + O(n \cdot m)$  → if  $n \sim m \rightarrow O(n^2)$

MergeSort Dynamic

Yer Karmaşıklığı:  $O(n) + O(n \cdot m) + O(n)$  → if  $n \sim m \rightarrow O(n^2)$   
Merge Dynamic Label

### **Requance Bağıntısı**

0.Sutun Reklam Numarası,1.Sutun Start Time, 2.Sutun Duration, 3.Sutun Value, 4.Sutun Profit, 'k' değeri çakışmayan son reklam indisidir. 0. Satır 0'lanır.

Matris[i][j] -----> Matris[0][i]= 0 i=0..3

Matris[i][0]= ReklamNo, Matris[i][1]= Start Time, Matris[i][2]= Duration, , Matris[i][3]= Value

Matris[i][3] = Profit[k] + Value[i] → if Profit[k] + Value[i] > Profit[i-1]

Matris[i][3] = Profit[i-1] → if Profit[k] + Value[i] < Profit[i-1]

### Uygulama:

j      Start Duration Value -----→Sıralama işlemi yapılacak.

1	5	3	3
2	9	4	7
3	11	6	9
4	4	7	5
5	1	3	2
6	2	5	3

j	START	DURATION	VALUE	PROFIT	AÇIKLAMA.....
1	1	3	2	2	Profit[0] + Value[1]>Profit[0] Bu yüzden Profit[1]=2
2	2	5	3	3	Profit[0] + Value[2]>Profit[1] Bu yüzden Profit[2]=3
3	5	3	3	5	Profit[1] + Value[3]>Profit[2] Bu yüzden Profit[3]=5
4	4	7	5	7	Profit[1] + Value[4]>Profit[3] Bu yüzden Profit[4]=7
5	9	4	7	12	Profit[3] + Value[5]>Profit[4] Bu yüzden Profit[5]=12
6	11	6	9	16	Profit[4] + Value[6]>Profit[5] Bu yüzden Profit[6]=16