Not# Teta simgesini bulmaadığım için onun yerine büyük Q kullandım. Büyük Q olan notasyonlarda teta kastetmek istedim.

1) Alan Turing zeki bir insan ve savaş zamanında geçiyor film. Filmde kararlı bir şekilde düşmanların kullandığı iletişim şifrelerini kırmak için bir makineye ihtiyaçları var ve bunun için bir ekip lazım. Alan Turing bunun için gönüllü oluyor.Bir çok zorluğa rağmen çalışmalarına devam ediyor.Insanlar tarafından pek sevilen bir insan değil kendisi. Bir ekip kurması isteniyor ve bunun sonrasında sınav yapıyor. Sınavda onu etkileyen bir kızı ekibine alıyor. Kimi zaman kendi ekibi bile ona inanmasada kendisi durmadan çalışmalarına devam ediyor ve sonunda başarıya ulaşıyorlar. Savaşı kazanıyorlar. Filmin sonuna doğru piskolojik bir çöküş yaşıyor ve hapis cezası gibi şeylerle karşılaşıyor ve ilaç tedavisine başlanıyor. Buna katlanamayıp kendisini zehirle öldürüyor.

2) $x_1(n) = \ 0.5x_1(\frac{n}{2}) + \frac{1}{n}$

a = 0.5

master teorem kuralları gereği a 1 den küçük olamaz bu yüzden x_1 master teorem ile çözülemez.

$x_2(n) = \ 3x_2(\frac{n}{4}) + nlogn4$

a=3 , b = 3 , f(n)=nlogn

nlogn > $n^{\log_4 3}$

3f($\frac{n}{4}$) < cf(n)

3$\frac{n}{4}$log$\frac{n}{4}$ < c.nlogn

C < 1

Q(nlogn)

$x_3(n) \ = \ 3x_3(\frac{n}{3}) + \frac{n}{2}$

a=3 , b=3 , f(n) = $\frac{n}{2}$ $\frac{n}{2}$ < $n^{\log_3 3}$

Q($n^{\log_3 3}$) = Q(n)

x_4(n) = 6 x_4 ($\frac{n}{3}$) + n^2logn

a = 6 , b=3 , f(n) = n^2logn n^2logn > $n^{\log_3 6}$

6f($\frac{n}{3}$) < cf(n) 6$\frac{n^2}{9}$log $\frac{n}{3}$ < c n^2logn c<1

Q(n^2logn)

x_5(n) = 4x_5($\frac{n}{2}$) + $\frac{n}{logn}$

a=4 , b= 2 , f(n) =$\frac{n}{logn}$ $\frac{n}{logn}$ < $n^{\log_2 2^n}$

Q($n^{\log_b a}$) = Q($n^{\log_2 4}$) = Q((n^2)

x_6(n) = $2^n x_6$ ($\frac{n}{2}$) + n^n

a=2^n , b = 2 , f(n) = n^n n^n > $n^{\log_2 2^n}$

2^n f($\frac{n}{2}$) < c.f(n)

2^n($\frac{n}{2}$)$^{\frac{n}{2}}$ < c.n^n c < 1

Q(n^n)

3 -a) T(n) = T(n-1) + 2n – 1 n ≥ 1 T(1) = 1

n = 5; T(5) = T(4) + 2*5 – 1 = 25 T(n) = n^2

T(4) = T(3) + 2*4 – 1 = 16

T(3) = T(2) + 2*3 – 1 = 9

T(2) = T(1) + 2*2 – 1 = 4

T(1) = 1

T(n-1) = $(n-1)^2$

n^2 = $(n-1)^2$+ 2n – 1

n^2 = n^2 – 2n + 1 + 2n – 1

n^2 = n^2

3-b) T(n) = T(n-1) +1 , eşitliğin sağ tarafındaki 1 asıl olan denklendeki çapma sayısı olduğu için her iterasyonda çarpma yapılacağından sonuçta kalan sayı toplam çarpın sayısı olacak.

T(n) = T(n-1) +1 k = n - 1

T(n-1) = T(n-2) +1

...

T(n-k-1) = T(n-k) +1

T(n-k) = 0 (T(1) = 1 ve burda herhangi bir çarpma işlemi uğgulanmamakta)

Bu eşitlikleri taraf tarafa topladığımızda

T(n) = (n-1)*1 yani n-1 tane çarpma işlemi olmaktadır.

3-c) T(n) = T(n-1) +3 , eşitliğin sağ tarafındaki 3 asıl olan denklendeki toplama ve çıkarma sayısı olduğu için her iterasyonda toplama ve çıkarma yapılacağından sonuçta kalan sayı toplam toplama ve çıkarma sayısı olacak.

T(n) = T(n-1) +3 k = n - 1

T(n-1) = T(n-2) +3

...

T(n-k-1) = T(n-k) +3

T(n-k) = 0 (T(1) = 1 ve burda herhangi bir toplama ve çıkarma işlemi uğgulanmamakta)

Bu eşitlikleri taraf tarafa topladığımızda

T(n) = (n-1)*3 yani 3n-3 tane çarpma işlemi olmaktadır.

4) Python kodu rar dosyasının içerisindedir.

5-a) Python kodu rar dosyasının içerisindedir.

5-b) Binary search tree gibi çalışıyor. Listeyi iki ayrı parçaya bölüp bunları karşılaştırıyor. Listenin küçük olan kısmını tekrar recursive olarak tekrar çağırıyor.Bu şekilde gide gide en sonunda tek bir eleman kalana kadar devam ediyor.Son kalan eleman ise çürük olan cevizin indexi oluyor.

Best case : O(1) , ilk seferde bulması

Worst case: Q(logn)

6-a) $T_1(n) = 3T_1\ (n-1) \quad for\ n > 1\ ,\ T_1(1) = 4$

n = 5; $T_1(5) = 3*T_1(4) = 324 = 4*3^4 = 4*3^{n-1}$

$T_1(4) = 3*T_1(3) = 108 = 4*3^3 = 4*3^{n-1}$

$T_1(3) = 3*T_1(2) = 36 = 4*3^2 = 4*3^{n-1}$

$T_1(2) = 3*T_1(1) = 12 = 4*3^1 = 4*3^{n-1}$

$T_1(1) = 4 = 4 = 4*3^0 = 4*3^{n-1}$

$T_1(n) = 4*3^{n-1}$

$T_2(n) = T_2\ (n-1) + n \quad for\ n > 1\ ,\ T_1(0) = 0$

n = 6; $T_2(6) = T_2(5) + 6 = 21 = \frac{6}{2} * 7 = \frac{n}{2} * (n+1)$

$T_2(5) = T_2(4) + 5 = 15 = \frac{5}{2} * 6 = \frac{n}{2} * (n+1)$

$T_2(4) = T_2(3) + 4 = 10 = \frac{4}{2} * 5 = \frac{n}{2} * (n+1)$

$T_2(3) = T_2(2) + 3 = 6 = \frac{3}{2} * 4 = \frac{n}{2} * (n+1)$

$T_2(2) = T_2(1) + 2 = 3 = \frac{2}{2} * 3 = \frac{n}{2} * (n+1)$

$T_2(1) = T_2(0) + 1 = 1 = \frac{1}{2} * 2 = \frac{n}{2} * (n+1)$

$T_2(0) = 0$ $= 0$

$T_2(n) = \frac{n}{2} * (n+1)$

$T_3(n) = T_3(\frac{n}{2}) + n$ n > 1 T(1) = 0 $n = 2^k$

k = 5; n = 32 ;	$T_3(32) = T_3(16) + 32$	= 62	=	2*n-2
k = 4; n = 16 ;	$T_3(16) = T_3(8) + 16$	= 30	=	2*n-2
k = 3; n = 8 ;	$T_3(8) = T_3(4) + 8$	= 16	=	2*n-2
k = 2; n = 4 ;	$T_3(4) = T_3(2) + 4$	= 6	=	2*n-2
k = 1; n = 2 ;	$T_3(2) = T_3(1) + 2$	= 2	=	2*n-2
	$T_3(1) = 0$	= 0	=	2*n-2

$T_3(n) = 2*n - 2$

6-b) $T_1(n) = 6T_1(n-1) - 9T_1(n-2)$, $T_1(0) = 1$, $T_1(1) = 6$

$a_n - 6a_{n-1} + 9_{n-2} = 0$

$r^n - 6r^{n-1} + 9r^{n-2} = 0$ tüm terimleri r^{n-2} ye bölüyoruz. En küçük olan olduğu için

$r^2 - 6r + 9 = 0$

(r-3)(r-3)=0 r=3,3

$a_n = x(3^n) + yn(3^n)$

$a_0 = 1 = x3^0 + y*0(3^n)$, x= 1

$a_1 = 1*(3^1) + y*1*(3^1) = 6$, 3+ 3y = 6 , y=1

$a_n = 3^n + n3^n$

$T_1(n) = 3^n + n3^n$

(kağıtta asıl denklemde n yerine 2 , 3 ,4 verince benim bulduğum sonuçta denklemler aynı sonuçlar veriyor. Buraya yazması çok uzun geldiği için böyle bıraktım.)

$T_2(n) = 5T_2(n-1) - 6T_2(n-2) + 7^n$ $a_n = 5a_{n-1} - 6a_{n-2} + 7^n$

$a_n = a_n^h + a_n^p$ $a_n - 5a_{n-1} + 6a_{n-2} = 7^n$

a_n^h bulmak için $a_n - 5a_{n-1} + 6a_{n-2} = 0$ (6-b şıkkındaki t1 deki dönüşümü yaptım yine)

r^2 – 5r +6 = 0 (dönüşmüş hali bu)

(r-2)(r-3) = 0

r = 3,2

a_n^h = x(3^n) + y(2^n)

a_n^p bulmak için a_n = 5a_{n-1} - 6a_{n-2} + 7^n Guess : a_n^p = A7^n

a_n gördüğüm yere A7^n yazdım.

A7^n - 5$A7^{n-1}$ + 6A7^{n-2} = 7^n her terimi 7^n e böldüm.

A – A$\frac{5}{7}$ + A$\frac{6}{49}$ = 1

49A – 35A – 6A = 49

5A = 49 A = $\frac{49}{5}$

a_n = x(3^n) + y(2^n) + $\frac{49}{5}$*7^n (youtube'da konu ile ilgili izlediğim videoda x yerine alfa y yerine beta kullanılıyordu. Yapması zor geldiği için onlar yerine x ve y kullandım.)

T_2(n) = x(3^n) + y(2^n) + $\frac{49}{5}$*7^n