1. Watch the film "The Imitation Game" which is based on the biography of Alan Turing. Write your thoughts about the film using an office program. It will at least occupy half of the A4 paper. (10 points)

Film Yapay Zeka'nın başlangıçını ve kriptografi gelişmesini anlatıyor. Kriptografinin ne kadar önemli olduğunu ve savaşların bile seyrini değiştirebiliceğini gösteriyor. Film 2. Dünya Savaşı sırasında Nazi Almanyası tarafından gizli mesajların şifrelenmesi için kullanılan Enigmayı anlatmıştır ve aynı zamanda film Enigmanın nasıl çözüldüğünüde anlatmaktadır. Enigmayı insan beyni ile çözülemeyeceğini anlayan Alan Matthison Turing bir makine yapmaya karar verir.

Ekip ile ilk başta anlaşamayan Alan sorunlar yaşamaktadır.Bence ekipteki bu durumlar onu negatif etkilemektedir. Özellikle yetkiyi aldıktan sonra iki dil bilimciyide kovması durumu daha kötü yönde etkiledi bence mantıklı bir karardı. Ekibi ile daha etkin çalışabilirdi. Ancak Alan kişisel tercihleri onun durumunu daha zora soktu. Ancak filmde önemli olan bir durum ise Alan'ın inancıydı. İnatla makinem çalışacak demesi ve bunun için çalışması birkez daha inançın ne kadar önemli olduğunu gözler önüne koydu.

Aynı zamanda Alan'ın amacı sadece enigmayı çözen bir makine değil aynı zamanda progranabilir bir makine yapmasaydı. Bu amaç uğrana uğraşması ve zaten zor olan şifre ile birlikte işleri bu denli zorlaştırması gerçekten hoşuma gitti. Enigmayı çözdüğünde esasında savaşı kazanmak için sadece şifreyi çözmenin yetmediğini. Aynı zamanda bu şifrenin çözüldüğünü almanlara fark ettirilmeden savaşı kazanması gerektiğini anladıklarında Alan'ın duygusalca düşünmemesi mantığıyla hareket etmesi caniceydi belki ama filmin güzel anlarındandı.

2. Solve following recurrence relations using master theorem. If any of the problems cannot be solved using master theorem, state that it can't be solved and explain the reason. (18 points)

$$X_1(n) = 0.5X_1\left(\frac{n}{2}\right) + \frac{1}{n}$$

a=0.5 b=2 a<1 olduğundan dolayı çözülemez. (Azalan)

$$X_2(n) = 3X_2\left(\frac{n}{4}\right) + n.logn$$

a=3 b=4 f(n) = n.logn -> polilogaritmik. Bundan dolayı özel durum handle edilir.

f(n) ϵ Θ(n.logn)

T(n) ϵ Θ(n.log$n$)

$$X_3(n) = 3X_3\left(\frac{n}{3}\right) + \frac{n}{2}$$

a=3 b=3 d=1

a=b^d -> 3 = 3^1

T(n) ϵ Θ($n^d.logn$)

T(n) ϵ Θ($n^1.logn$) => Θ($n.logn$)

$X_4(n) = 6X_4\left(\frac{n}{3}\right) + n^2$.logn

a=6 b=3 f(n) = n^2.logn -> logaritmik olduğundan.

f(n) ϵ Θ(n^2.logn)

T(n) ϵ Θ(n^2.logn)

$$X_5(n) = 4X_5\left(\frac{n}{2}\right) + \frac{n}{logn}$$

a=4 b=2 f(n) =n/logn -> logaritmik olduğundan.

f(n) ϵ Θ($n^{log_2 4}$ $.log^{-1}n$)

T(n) ϵ Θ(n^2)

$$X_6(n) = 2^n X_6\left(\frac{n}{2}\right) + n^n$$

a=2^n => sabit olmadığından çözülemez.

3.Consider the following algorithm implemented in python: (18 points)

```
def chocolateAlgorithm(n):
    #Input is a positive integer n
    if n==1:
        return 1
    else:
        return chocolateAlgorithm(n-1) + 2 * n -1
```

a-) Set up a recurrence relation for this function's values and solve it to determine what this algorithm computes.

T(n)=T(n-1)+2n-1 for n>1, T(1)=1

T(1) =1
T(2) =T(1)+3=4
T(3) =T(2)+5=9
T(4) =T(3)+7=16
T(5) =T(4)+9=25
Fonksiyon verilen inputun karesini hesaplar.
Bu nedenle, T(n)=n^2 olur. Verilen ilk değere ve tekrar fonksiyonlarında denersek.
T(n-1)+2n-1 = $(n-1)^2 + 2n - 1$ = n^2
T(1)= $1^2 = 1$

b-) Set up a recurrence relation for the number of multiplications made by this algorithm and solve it.

Z(n) = Z(n-1)+1 for n>1 Z(1)=0

Z(1) = 0
Z(2) = 1
Z(3) = 2
Z(4) = 3
……….
Z(n)=n-1 varsayarsak, Z(n-1) = n-2
n-1 = n-2+1, eşitlik durumu sağlanır. Ve doğrudur.

c-) Set up a recurrence relation for the number of additions/subtractions made by this algorithm and solve it

Toplamlara ve çıkarmalara Y(n) dersek;
Recurrence formulumuz(iki toplama ve bir çıkarmadan,ve tekrar recursive kol yapmasından);
Y(n) = Y(n-1)+3 for n>1 Y(1)=0

Y(1) = 0
Y(2) = 3
Y(3) = 6
Y(4) = 9
…….
Y(n) = 3(n-1) varsayarsak;
Y(n-1) = 3(n-2) => Y(n) =? Y(n-1)
3(n-1) = 3(n-2)+3
Bu durumda sağlanır ve doğrudur.

5. (18 points) Consider the problem of finding rotten walnut. You have n walnuts and the weights of all walnuts are equal except the rotten one which is lighter than the others. Your input will be a python list of positive integers which indicates the weight of all walnuts; for example: [1 1 1 1 1 0.5 1 1 1]. Your output (return value of function) will be the index of the rotten walnut. To find the rotten walnut, you will use a pair of scales. The Python function which compares two set of walnuts is:

b) Explain your algorithm. Show this algorithm's number of operations in terms of input array size (n) and complexity using the asymptotic notations for best and worst cases.

Öncelikle gelen arrayi eleman sayısının çift olması ve tek olması olarak ayırdım, ondan sonra hocanında verdiği fonksiyonu kullanarak eğer compareScales fonksiyonu 1 döndürüyorsa arrayın solundan eğer -1 veriyorsa arrayın sağından aramaya devam ettim. (Tabi ortasından bölüp tekrar yolluyorum. BinarySearch gibi.)

n elemanlı array için örnek verirsek,
[1,1,1,1,1,1,0.5,1,1,1]
    [1,0.5,1,1,1]
    [1,0.5,1]   = > bu aşamada index return eder.

Best-Case: Eğer eleman sayımız iki ise (2 elemanlı bir array) best-case durumumuz O(1) dir.
[1,0.5] veya [0.5,1] bu tip data da yazdığım kod O(1) karmaşıklıkta çalışmaktadır.

Worst-Case: n elemanlı bir array da her adımda array elamanı sayısı yarıya düşeceğinden, sürekli her bir sonraki kolda elaman sayısı yarıya düşmektedir. (c olarak bahsettiğim işlemler constant zamandaki sabit sayılar gibi düşünebiliriz.)

$$T(n) = T\left(\frac{n}{2}\right) + c$$
$$T(n) = T\left(\frac{n}{4}\right) + c + c$$
$$T(n) = T\left(\frac{n}{8}\right) + c + c + c$$
$$T(n) = T\left(\frac{n}{2^k}\right) + c * k$$
$$T(n) = logn + C$$
$$T(n) = O(logn)$$
$$T(n) \epsilon O(logn)$$

Genel olarak baktığımızda arrayimizin sürekli size ı yarıya düşeceğinden karmaşıklık analizi için O(logn) diyebiliriz. (2 elamanlı best case durumu çok istisnadır.)

$$T(n) = T\left(\frac{n}{2}\right) + c$$
$$T(n) = T\left(\frac{n}{4}\right) + c + c$$
$$T(n) = T\left(\frac{n}{8}\right) + c + c + c$$
$$T(n) = T\left(\frac{n}{2^k}\right) + c * k$$
$$T(n) = logn + C$$
$$T(n) = O(logn)$$
$$T(n) \epsilon O(logn)$$

6. (18 points) Solve the following recurrence relations,
a) Using forward/backward substitution:

$T_1(n) = 3T_1(n-1)$ for n>1, $T_1(1) = 4$

$T_1(1) = 4$
$T_1(2) = 3.4$
$T_1(3) = 3.3.4$
$T_1(4) = 3.3.3.4$
...
...
$T_1(n) = 3^{n-1}.4$

Varsayarsak; $T_1(n) = 3^{n-1}.4$
$T_1(n-1) = 3^{n-2}.4$

$3^{n-1}.4 =? \ 3.3^{n-2}.4$
Eşit olduğundan, $3^{n-1}.4 = 3^{n-1}.4$
$\Theta(3^n)$

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

$T_2(0) = 0$
$T_2(1) = 1$
$T_2(2) = 1 + 2$
$T_2(3) = 1 + 2 + 3$
...
...
$T_2(n) = 1 + 2 + 3 + \cdots + n$

$T_2(n) = 1 + 2 + 3 + \cdots + n$ olduğundan,

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

$$T_2(n) \ \epsilon \ \Theta(n^2)$$

$T_3(n) = T_3\left(\frac{n}{2}\right) + n$ for n>1, $T_3(1) = 0$ (Solve for n=2^k)

$T_3(1)$ =0
$T_3(2)$ =0+2
$T_3(4)$ =0+2+4
$T_3(8)$ =0+2+4+8
...
...
$T_3(2^k)$ =0+2+4+8+...+2^k

$$\sum_{i=0}^{k} 2^i$$

$\int_0^n 2^x \, dx$<= f(x) <= $\int_1^{n+1} 2^x \, dx$

$$T_3(n) \ \epsilon \ \Theta(2^n)$$

b-) Using the properties of linear homogeneous/inhomogeneous equations:

T(n) = 6T(n-1)-9T(n-2) T(0)=1, T(1)=6
Karakteristik Denklemini yazarsak;

$$r^2 - 6r + 9 = 0$$
$$(r-3)^2 = 0$$
r=3 çift katlı kök.
$$a_n = (c1 + c2.n)3^n$$
$$a_0 = (c1 + c2.0)3^0 \ \text{--} \ a_0 = c1 = 1$$
$$a_1 = (c1 + c2.1)3^1 \ \text{--} \ a_0 = 3c1 + 3c2 = 6$$
$$c1 = 1, c2 = 1$$
So that, $a_n = (1 + 1.n)3^n$

T(n) = 5T(n-1)-6T(n-2)+7^n

$$a_n - 5a_{n-1} + 6a_{n-2} = 0$$
$$a_n = a_n^h + a_n^p$$

a_n^h --- >

$$a_n - 5a_{n-1} + 6a_{n-2} = 0$$
$$r^2 - 5r + 6 = 0$$
$$(r-2)(r-3) = 0$$
$$a_n^h = x(2)^n + y(3)^n$$

a_n^p --- >

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

a_n => A. 7^n varsayalım.

$$A7^n - 5.A.7^{n-1} + 6.A.7^{n-2} = 7^n$$
$$49A - 35A + 6A = 49$$
$$A = \frac{49}{20}$$
$$= \frac{49}{20}7^n$$

$$a_n = a_n^h + a_n^p$$
$$a_n = x(2)^n + y(3)^n + \frac{49}{20}.7^n$$