1)

Well, this is hard to describe what I think and feel right now. After watching the film, I do a little extra research about Alan Turing. I always heard his name, Turing Test and Turing Machine but never ask the question “Why I hear his name all the time, what are these things?” to neither myself nor Google. First thing come to my mind, after looking Turing’s works, is the Singularity hypothesis. If we continue with Imitation Game and think about Bombe, it was just trying to get a meaningful message from an encrypted text by trying each possible combination of Enigma to get the key used for encryption. I can’t figure out how intelligent Bombe or his other machines are since a human can try each combination to find a valid key. They do the same work. But since there are between 10^19 and 10^22 [1] (nearly 150 trillion, or 1-5-9 with eighteen 0 behind according to the Imitation game) ten men controlling a configuration in a minute would take twenty million years (Twenty-eight million years according to referenced data) to find correct settings for Enigma Machine. It makes finding the key impossible due to German’s security policy that requires change of key each day. Even if the key was found for that day, you need to redo all controls for next day. But a brain and hands working faster would reach the result faster. This isn’t just for Enigma Machine or decrypting a message. ENIAC which considered amongst the first computers [2] were doing mathematical calculations for physics formulas which humans are capable of too. But if we consider the meaning of power, which is the rate of doing work in unit time [3], that machines build during the second World War are still powerful than a human brain. Machines back then were designed to do a specific job or work like solving equations or ciphered texts etc. But they were also generalized for that specific job. You did not need a second Bombe to solve next day’s key for Enigma Machine. It was actually capable to think on its own mind which only understands from solving puzzles. Then humanity started to expand the minds of computers to make them more generic. Today’s computers can both solve mathematical equations and decrypt a message and helps me to search things on other computers/machines, watch films or play games. A computer takes inputs, processes the input and produces an output. We can consider it like a computer listens what you say to it, thinks about what to do and does what needs to be done. This is only limited by what we taught to it. An artificial intelligence (from what I understand) is simply considers all the available option it has, to find best case for next step. If we manage to teach how to analyse problems and teach how to find an algorithm that solves the problem to a machine, then won’t we simply build an AI, that thinks and acts on its own? An algorithm that designs other algorithms might sound impossible for now. But if we think about other things considered as impossible back then and what humanity has come in two thousand years, all I can say is “Why not?”. I hope, if we ever accomplish this, it won’t result in bad ways.

References:

1- https://en.wikipedia.org/wiki/Cryptanalysis_of_the_Enigma#Security_properties
2- https://en.wikipedia.org/wiki/ENIAC
3- https://en.wikipedia.org/wiki/Power_(physics)

Introduction to Algorithm Design HW2

Deniz Can Erdem
Yilmaz
151044001

2) Master Theorem*:

If $T(n) = a\,T(n/b) + f(n)$ where $a>0$, $b>1$ then

case 1: if $f(n) \in O(n^c)$, where $c < \log_b a$ $(b^c < a)$
then $T(n) \in \theta(n^{\log_b a})$

case 2: if $f(n) \in \theta(n^c \cdot \log^k n)$, where $c = \log_b a$ and $k >= 0$ $(b^c = a)$
then $T(n) \in \theta(n^c \cdot \log^{k+1} n)$

case 3: if $f(n) \in \Omega(n^c)$, where $c > \log_b a$ and $a.f(n/b) \leq k.f(n)$, $k<1$
then $T(n) \in \theta(f(n))$ $(b^c > a)$

* Theorem taken from video:
https://youtu.be/mSX8jFgTC=0

So;

a) $x_1(n) = 0{,}5\, x_1\left(\frac{n}{2}\right) + \frac{1}{n}$

$a = 0{,}5$
$b = 2$
$f(n) = n^{-1}$

$a > 0$, but it isn't integer
Can't be solved

b) $x_2(n) = 3x_2\left(\frac{n}{4}\right) + n.\log n$

$a = 3$
$b = 4$
$f(n) = n \log n$
$c = 1$

$b^c \ ?\ a$
$4^1 \ ?\ 3$
$4 > 3$

$\rightarrow n \log n \in \Omega(n)$
and $\frac{3n}{4}.\log\frac{n}{4} \leq \frac{3}{4} f(n)$
$\searrow k$

So $x_2(n) \in \theta(n.\log n)$
(case 3)

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

$a = 3$
$b = 3$
$f(n) = \frac{n}{2}$
$c = 1$

$b^c \ ?\ a$
$3^1 \ ?\ 3$
$3 = 3$

$\rightarrow$ for $k=0$
$f(n) \in \theta(n^c \log^k n)$
so $x_3(n) \in \theta(n \log n)$
(case 2)

d) $x_4(n) = 6x_4\left(\frac{n}{3}\right) + n^2 \log n$

$a = 6$
$b = 3$
$f(n) = n^2 \log n$
$c = 2$

$b^c \ ?\ a$
$3^2 \ ?\ 6$
$9 > 6$

$\Rightarrow n^2 \log n \in \Omega(n^c)$
and $\frac{6n^2}{9}.\log\frac{n}{3} \leq \frac{2}{3} \log n$
$\searrow k$

so $x_4(n) \in \theta(n^2 \log n)$
(case 3)

e) $x_5(n) = 4\,x_5\left(\frac{n}{2}\right) + \frac{n}{\log n}$

$a = 4$
$b = 2$
$f(n) = \frac{n}{\log n}$
$c = 1$

$b^c \ ?\ a$
$2^1 \ ?\ 4$
$2 < 4$

$\rightarrow \frac{n}{\log n} \in O(n^c)$

so $x_5(n) \in \theta(n^2)$
(case 1)

f) $x_6(n) = 2^n\, x_6\left(\frac{n}{2}\right) + n^n$

$a = ?$
$b = 2$
$f(n) = n^n$
$c = 1$

a is not an integer
Can't be solved

3) def chocolateAlgorithm(n):

```
    if n==1:
        return 1
    else:
        return chocolateAlgorithm(n-1) + 2*n - 1
```

a) Function finds the square of given n value

$T(n) = T(n-1) + 2n - 1$ & $T(1) = 1$ so;

$$\left.\begin{aligned} T(n) &= T(n-1) + 2n - 1 \\ T(n-1) &= T(n-2) + 2n - 3 \\ T(n-2) &= T(n-3) + 2n - 5 \\ &\vdots \\ T(2) &= T(1) + 3 \\ T(1) &= 1 \end{aligned}\right\} n$$

$$T(n) = (2n-1) + (2n-3) + (2n-5) + \dots + 3 + 1 = \sum_{i=1}^{n} (2i - 1)$$

$$= \sum_{i=1}^{n} 2i - \sum_{i=1}^{n} 1$$

$$= 2\sum_{i=1}^{n} i - n$$

$$= 2 \cdot \frac{n(n+1)}{2} - n = n^2 + n - n = n^2$$

b) There is only one multiplication on each recursive call line. So recurrence relation with numbers of multiplication must be;

$T(n) = T(n-1) + 1$ and

$$\left.\begin{aligned} T(n) &= T(\cancel{n-1}) + 1 \\ T(\cancel{n-1}) &= T(\cancel{n-2}) + 1 \\ T(\cancel{n-2}) &= T(\cancel{n-3}) + 1 \\ &\vdots \\ T(\cancel{2}) &= T(\cancel{1}) + 1 \end{aligned}\right\} 1*(n-1)$$

$T(\cancel{1}) = 0$ (no multiplication)

$T(n) = n - 1$

c) There are two add/subs for recursive call. So recurrence relation must be;

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

$$\left.\begin{aligned} T(n) &= T(\cancel{n-1}) + 2 \\ T(\cancel{n-1}) &= T(\cancel{n-2}) + 2 \\ T(\cancel{n-2}) &= T(\cancel{n-3}) + 2 \\ &\vdots \\ T(\cancel{2}) &= T(\cancel{2-1}) + 2 \end{aligned}\right\} n-1$$

$T(\cancel{1}) = 0$ (no add or sub)

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

$= 2n - 2$

6)

a.1) $T(1) = 4$
$T(2) = 12$
$T(3) = 36$
$T(4) = 108$
$\vdots$

can it be $(3^{n-1} . 4)$?

if $T(n) = 3^{n-1} . 4$ then

$$\underbrace{3^{n-1} . 4}_{T(n)} = \underset{3}{3}(\underbrace{3^{n-2} 4}_{T(n-1)})$$

$$3^{n-1} . 4 = 3 . 3^{n-2} . 4$$

$$3^{n-1} . 4 = 3^{n-1} . 4 \checkmark$$

so $T(n) = 3^{n-1} 4$

a.2) $T(0) = 0$
$T(1) = 1$
$T(2) = 3$
$T(3) = 6$
$T(4) = 10$

can it be $\left(\frac{n.(n+1)}{2}\right)$?

if $T(n) = \frac{n^2+n}{2}$ then

$$\frac{n^2+n}{2} = \frac{(n-1).n}{2} + n$$

$$n^2+n = n^2 - n + 2n$$

$$n^2+n = n^2+n \checkmark$$

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

a.3) $T(1) = 0$
$T(2) = 2$
$T(4) = 6$
$T(8) = 14$

can it be $(2n-2)$?

if $T(n) = 2n-2$ then

$$2n-2 = \frac{2n}{2} - 2 + n$$

$$2n - 2 = 2n - 2 \checkmark$$

so $T(n) = 2n-2$

---

b.1) $f(n) = 6f(n-1) - 9f(n-2)$ $\quad f(0) = 1$, $f(1) = 6$

Char. equation $\rightarrow x^2 = 6x - 9$
$x^2 - 6x + 9 = 0$
$(x-3)^2 = 0$
$x_1 = x_2 = 3$

$f(n) = c_1 3^n + c_2 n 3^n$

$f(0) = c_1 = 1$
$f(1) = 3c_1 + 3c_2 = 6$

$c_1 = 1$
$c_2 = 1$

$f(n) = 3^n + 3^n . n$

$f(n) = 3^n(n+1)$

b.2) $f(n) = 5f(n-1) - 6f(n-2) + 7^n$

Char equation $\rightarrow x^2 = 5x - 6$
$x^2 - 5x + 6 = 0$
$(x-2)(x-3) = 0$
$x_1 = 2 \quad x_2 = 3$

for $f(n) = c_3 . 7^n$

$$c_3 . 7^n = \frac{5c_3 . 7^n}{7} - \frac{6c_3 . 7^n}{49} + 7^n$$

$$49c_3 . 7^n = 35 c_3 7^n - 6c_3 . 7^n + 7^{n+2}$$

$$20c_3 . 7^n = 7^n . 49 \rightarrow c_3 = \frac{49}{20}$$

$$f(n) = c_1 2^n + c_2 3^n + \frac{7^{2+n}}{20}$$