Question 1

Turing starts by posing a challenge, if a computer think then it should convince an interrogator that it is human. He also questions the definition of a machine, if we cannot describe its function, is it a machine?

He then goes on to define the fundamental instructions a computer needs: Store, Execute (operations), Control (instruction cache and opcode) as well as defining the branch (if zero) operation "If position 4505 contains 0 obey next the instruction stored in 6707...". It is then asserted as fact if we build computers in this way, computers should thus be able to mimic humans (closely). However he also says that electricity is unnecessary for thought but we could theoretically build a mechanical machine that would be able to "think".
He goes on to define the original question "Can machines think?" to "Are there discrete state machines which would do well?" which is a much less ambiguous question. He also explores where computers will fail in the interrogator scenario, citing "thought" as the primary example (e.g. "what do you think of this musician?"). He counters various arguments to do with the analysis of specialised machines and how machines in general would never be able to do X, where X is a set of inductive/opinionated judgements, his main counter arguments are absurdity/farce/logical-errors. He also evaluates the human CNS as a continuous state system, and compares machines and humans, expecting it impossible to accurately model the human nervous system. He counters the idea posed by Lady Lovelace (machines can only do what we tell them to) by asking to model the mind of a child and educating that mind would produce that of an adult mind.

To conclude turning give his opinion of the future of machine, machines will think and will compete with humans on an intellectual level.

Question 2:

Question 2.a

The state space depends on two elements:

- The node you are at: $n\ different\ value$
- If the node h has been visited $2\ different\ value$

Then the state space is $2 * n$ and $24 = 2 * 12$ is this case

Question 2.b

Similarly for this scenario we still have $n\ node$ but this time we have k node to visit.

Then for $i \in [1, k]$ we have the node n_i to visit which gives us two different states.

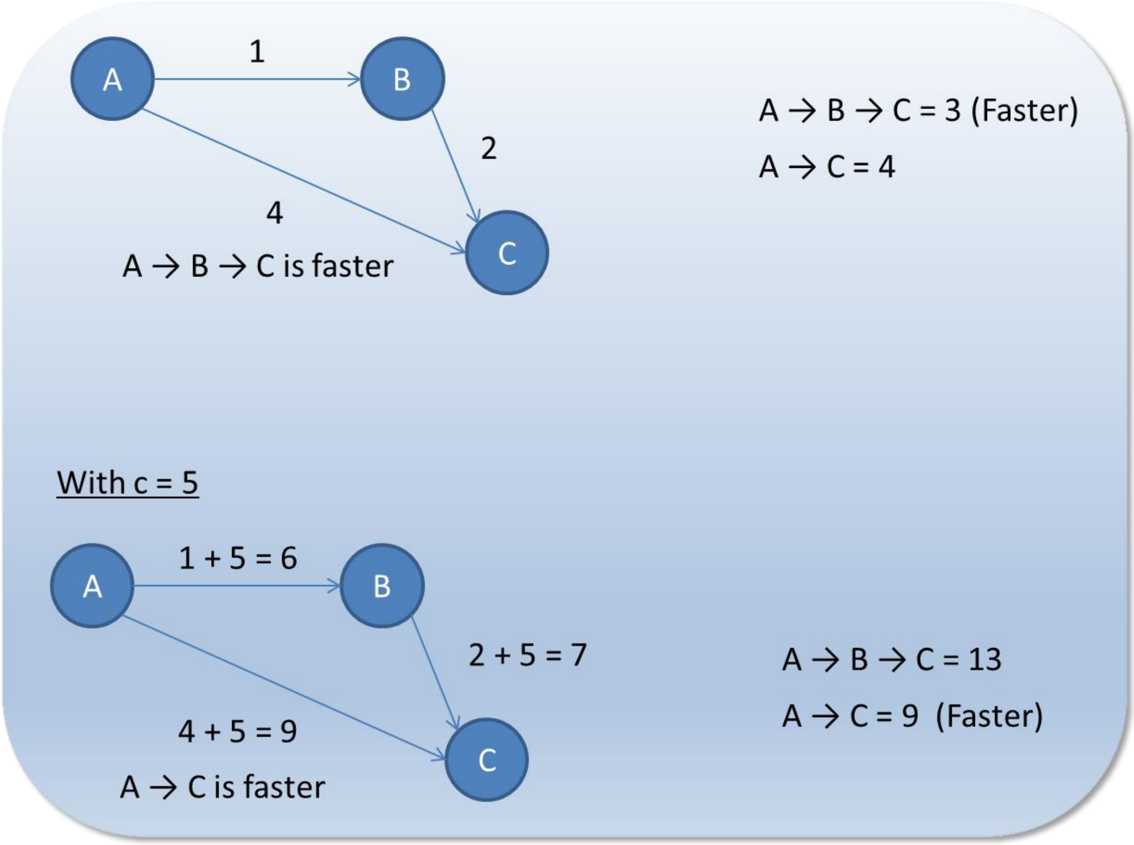
Thus we get the state space to be $2^k * n$

Question 2.c

Taking the previously found formula and putting $k = n$ as the number of node is n and all the node needs to be visited, we get $2^n * n$ for the state space size

Question 3

The following example contradicts the proposition:



With the first graph we see that the fastest path from A to C is going through B however we notice that by adding 5 on all edges going directly from A to C is faster.

Question 4

$$h(n) < h^*(n) + \epsilon$$

Let n^*be the optimal node and n_g the node obtained using $h(n)$
Let k be a node then

$$f(k) = g(k) + h^*(k) < g(n^*)$$

$$f(k) = g(k) + h(k) < g(k) + h^*(k) + \epsilon < g(n^*) + \epsilon$$

As $h(n)$ overestimate de path then we have

$$g(n_g) \leq f(k)$$

thus by transitivity

$$g(n_g) \leq \ g(n^*) + \epsilon$$

Question 5

Question 5.a

Each vehicle can be place on the grid but there cannot be two at the same place. Which means that the first one have $n * n$ different position, the second one only have $n * n - 1$ left and the nth has $n * n - n + 1$.

Then we get the following formula for the state space:

$$State\ space = \frac{(n^2)!}{n^2 - n}$$

Question 5.b

Each vehicle can move to 4 different positions (Up, Right, Down, Left) then with n vehicle we get the following formula:

$$Branching\ factor = 4^n$$

Question 5.c

The vehicle is at position $(x_i\ , y_i)$ and needs to go to position $(n\ , n - y_i + 1)$. Then the vehicle has to move $n - \ x_i$ in the x axis and $n - y_i + 1 - \ y_i$ in the y axis. Thus we get the following formula:

$$h_1 = 2n - x_i - 2y_i + 1$$

Question 5.d

i) **Is not admissible**, because it implies that only one piece can move at a time; no concurrency. We know this to be false.
We can get pieces to move concurrently and thus this heuristic is at best a large upper bound to the actual number of steps.
It is not admissible because it over-estimates not under-estimates

ii) **Is admissible,** because it implies that the total number of steps is equal to the number of steps it takes to move the piece
that takes the largest number of steps, i.e. the path with the largest number of moves. In practice, it would probably take
a few more steps, because this piece may need to wait for another, but it is definitely more suitable than the other heuristics.
It is admissible because it under-estimates the number of steps, the longest path would still probably be blocked and have to wait
for another piece to move, and thus is a lower bound.

iii) **Is admissible** because it under-estimates the number of moves it takes. It will clearly take more moves than the minimum, otherwise all
pieces would have to move the same distance or travel at different speeds, which is untrue.