Question 1:
(a)

	Start10	Start20	Start27	Start35	Start43
UCS	2565	Mem	Mem	Mem	Mem
IDS	2407	5297410	Time	Time	Time
A∗	33	915	1873	Mem	Mem
IDA∗	29	952	2237	215612	2884650

(b)
For the UCS and IDS, the time complexity of them are similar. But the space complexity of UCS is much higher than that of IDS.

For the A* and IDA*, both the time complexity of them are much less than those of UCS and IDS. But the space complexity of IDA* is less than A*.

Therefore, IDA* is the best path search algorithm of these four algorithms.

Question 2:
(a)
?- start49(S), showpos(S), h(S,H).

MBDC
LAKH
JFEI
 ONG
S = [4/1, 2/2, 1/2, 1/4, 1/3, 3/3, 3/2, 4/4, ... / ...|...],
H = 25.

?- start51(S), showpos(S), h(S,H).

GKJI
 MNC
EOHA
FBLD
S = [2/1, 3/4, 4/2, 2/4, 4/4, 3/1, 4/1, 1/1, ... / ...|...],
H = 43.

(b)
The number of nodes is 551168.

(c)
The issue is caused by the heuristic function. As is shown above, heuristic function estimates the value h(n) of start49 too lower than actual circumstance. It guides the search incorrectly and leads to the node’s expansion.

Question 3:
(a)

	Start49		Start60		Start64	
IDA∗	49	178880187	60	321252368	64	1209086782
1.2	51	988332	62	230861	66	431033
1.4	57	311704	82	4432	94	190278
Greedy	**133**	**5237**	**166**	**1617**	**184**	**2174**

(b)

```
% F1 is G1 + H1.
 F1 is 0.8 * G1 + 1.2 * H1,
```

(c)

	Start49		Start60		Start64	
IDA∗	49	178880187	60	321252368	64	1209086782
1.2	**51**	**988332**	**62**	**230861**	**66**	**431033**
1.4	**57**	**311704**	**82**	**4432**	**94**	**190278**
Greedy	133	5237	166	1617	184	2174

(d)
According to the objective function of heuristic path algorithm, f(n) = (2 – w)*g(n) + w*h(n).

When w = 1, it reduces to A* Search. IDA* Search has the shortest path but searching the most nodes.

The more w is, the quality is poorer, but the time spent is less. As is shown above, the length of path when w = 1.4 is more than those when w = 1.2. And the total number of states when w = 1.4 is less than those when w = 1.2.

When w = 2, it reduces to Greedy search which is the fastest search of these four algorithms. But its quality is the poorest because it has the longest path.

Question 4:
(a)
We may use Manhattan-distance heuristic.
h(x, y, xG, yG) = |x – xG| + |y – yG|

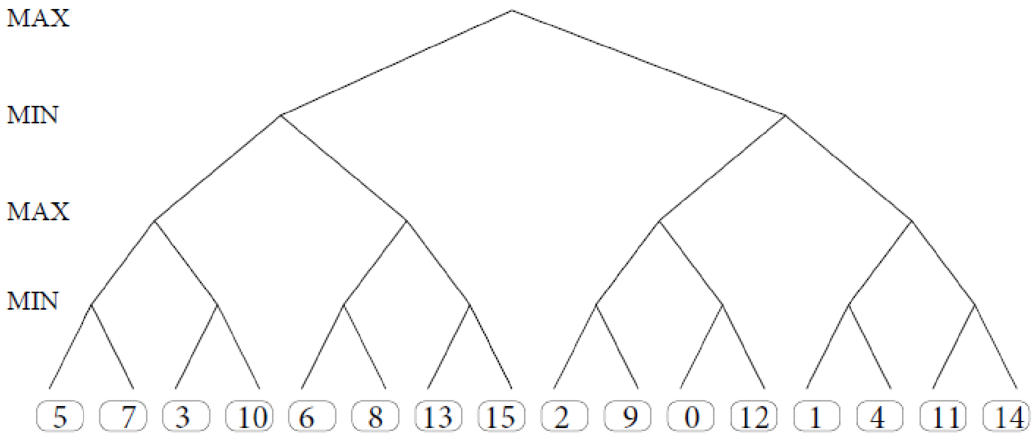
(b)
(i) No, it is not admissible.

The definition of admissible is that for every n, it satisfies h(n) ≤ h*(n).
For example, hSLD(0, 0, 2, 2) = 2√2, h*(0, 0, 2, 2) = 2.
hSLD(0, 0, 2, 2) > h*(0, 0, 2, 2), so the Straight-Line-Distance heuristic is not admissible.

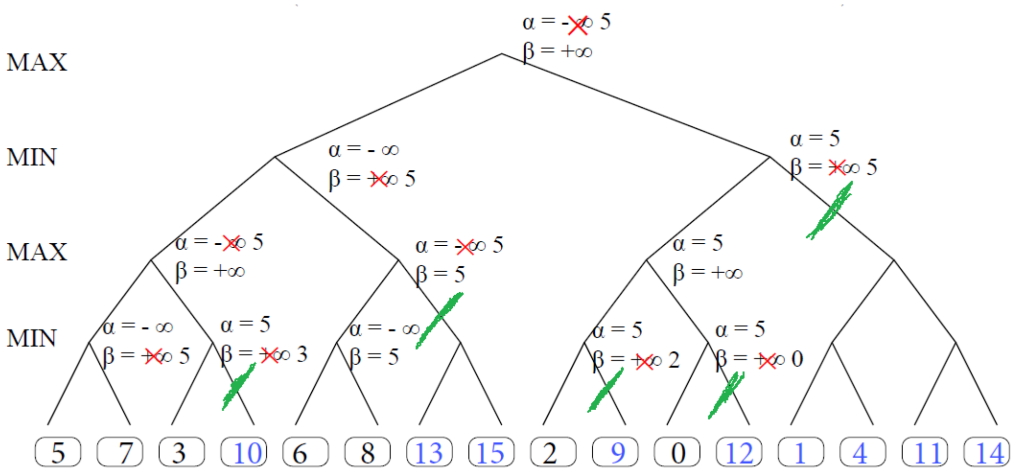
(ii) No, it is not admissible.
The definition of admissible is that for every n, it satisfies h(n) ≤ h*(n).
For example, h(0, 0, 2, 2) = 4, h*(0, 0, 2, 2) = 2.
h(0, 0, 2, 2) > h*(0, 0, 2, 2), so my heuristic from part (a) is not admissible.

(iii)
To satisfy this 8-direction movement, we may use diagonal heuristic.
h(x, y, xG, yG) = max(|x – xG|, |y – yG|)

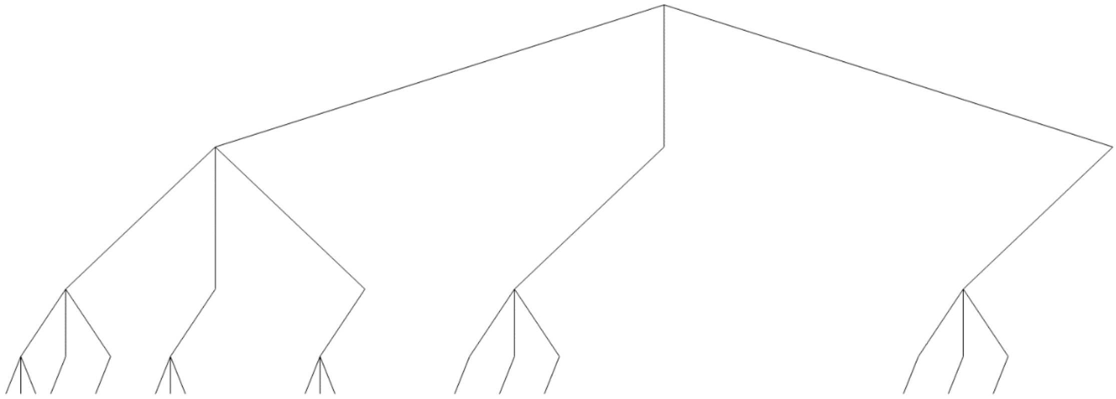
Question 5:
(a)



(b)



The nodes evaluated are coloured black, while the nodes pruned are coloured blue. And the green line indicates the pruning operation.
(c)



As the pruned tree shown in the diagram above, only 17 original leaves are evaluated.

(d) If the best move is always examined first in alpha-beta search, on each “max” layer, we need to consider only the far-left extension of each node. While on each “min” layer, we need to consider all extensions of each node.

We may use “b” to indicate branching factor and “d” to indicate depth of the tree. T(n) = 1 in “max” layer and T(n) = b in “min” layer.

T(n) = b * 1 * b * 1 …… = O(b^(d/2))

Thus, the time complexity is O(b^(d/2)).