Assignment name: assignment 2

Question1:

(a):

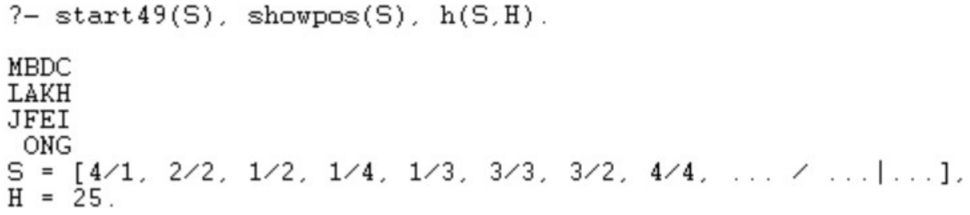
	Start10	Start20	Start27	Start35	Start43
UCS	2565	Mem	Mem	Mem	Mem
IDS	2407	5297410	Time	Time	Time
A*	33	915	1873	Mem	Mem
IDA*	29	952	2237	215612	2884650

(b):

As the table showed, IDA* is the most efficient algorithm of the four algorithms because it saves time and space. Followed by the A*, it is also an efficient algorithm although the Start35 and Start43 are out of memory meaning out of space which is also more efficient than IDS and UCS. Then, IDS is less efficient way due to it is out of time. Finally, the UCS is the least efficient algorithm since it is usually out of memory, for example, it is out of space in Start20, Start27, Start35 and Start43.

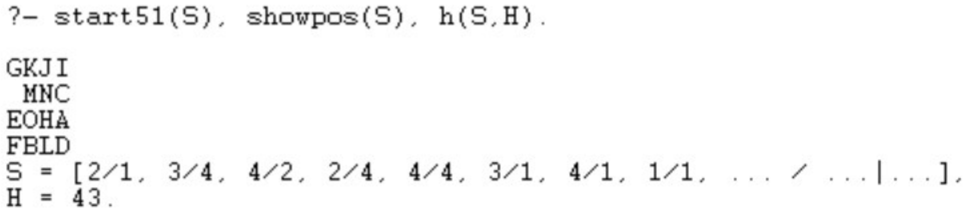
Question2:

(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.
```

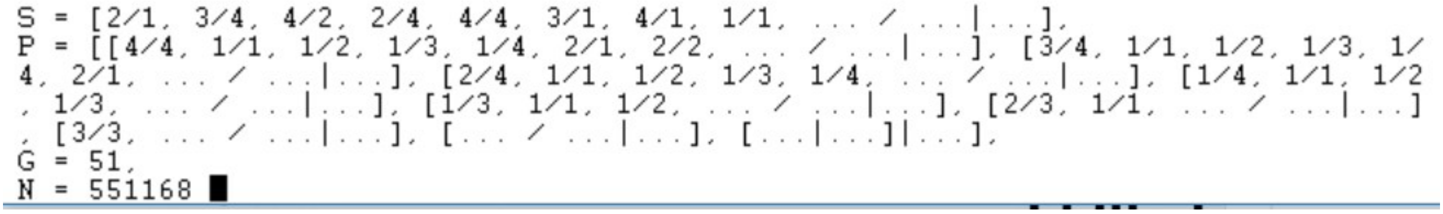


```
?- start51(S), showpos(S), h(S,H).

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

(b):

551168



```
S = [2/1, 3/4, 4/2, 2/4, 4/4, 3/1, 4/1, 1/1, ... / ...|...],
P = [[4/4, 1/1, 1/2, 1/3, 1/4, 2/1, 2/2, ... / ...|...], [3/4, 1/1, 1/2, 1/3, 1/
4, 2/1, ... / ...|...], [2/4, 1/1, 1/2, 1/3, 1/4, ... / ...|...], [1/4, 1/1, 1/2
, 1/3, ... / ...|...], [1/3, 1/1, 1/2, ... / ...|...], [2/3, 1/1, ... / ...|...]
, [3/3, ... / ...|...], [... / ...|...], [...|...]|...],
G = 51,
N = 551168 █
```

(c):

As we can see from h of Start49 and Start51, the h of Start49 is only 25 which is underestimate, compared with Start51 which is 43 of h. Start49 may take some wrong ways which results in the number of nodes bigger than that of Start51.

Question3:

(a):

	Start49		Start60		Start64	
IDA*	49	178880187	60	321252368	64	1209086782
1.2 1.4						
Greedy	133	5237	166	1617	184	2174

(b):

the section of code that was changed: F1 is G1 + H1, the replacement code: F1 is (2-1.2)*G1 + 1.2*H1

(c):

	Start49		Start60		Start64	
IDA*	49	178880187	60	321252368	64	1209086782

1.2 1.4	51 57	988332 311704	62 82	230861 3673	66 94	431033 188917
Greedy	133	5237	166	1617	184	2174

(d):

When the function IDA* change to Greedy, the speed will increase, but the quality will be decreased. Additionally, the optimal solution is IDA*, but the Greedy is not optimal solution.

Question4:

(a):

$h(x, y, x_G, y_G) = |x- x_G |+|y- y_G |$

(b):

(i)

It is not admissible $h_{SLD}(0, 0, 1, 1) =\sqrt{2}$ $h_{ad}(0, 0, 1, 1) = 1$

because $h_{SLD} > h$, so, it violates the principle of admissible.

(ii):

It is not admissible $h(0, 0, 1, 1) = |1-0|+|1-0| = 2$ $h_{ad}(0, 0, 1, 1)$

$= 1$ because $h > h_{ad}$, so, it also violates the principle of

admissible.

(iii):

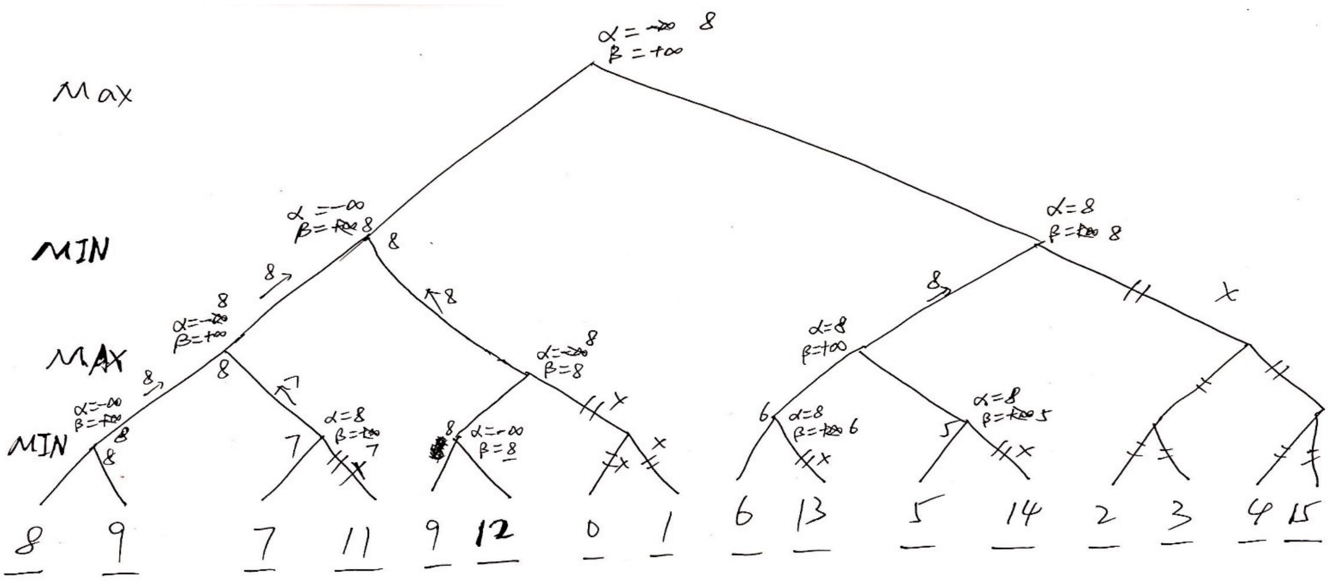
$h(x, y, x_G, y_G) = \max(|x- x_G |,|y- y_G |)$

$h(0, 0, 1, 1) = 1$ $h_{ad}(0, 0, 1, 1) = 1$

because $h = h_{ad,}$ so, it is admissible.

Question5:

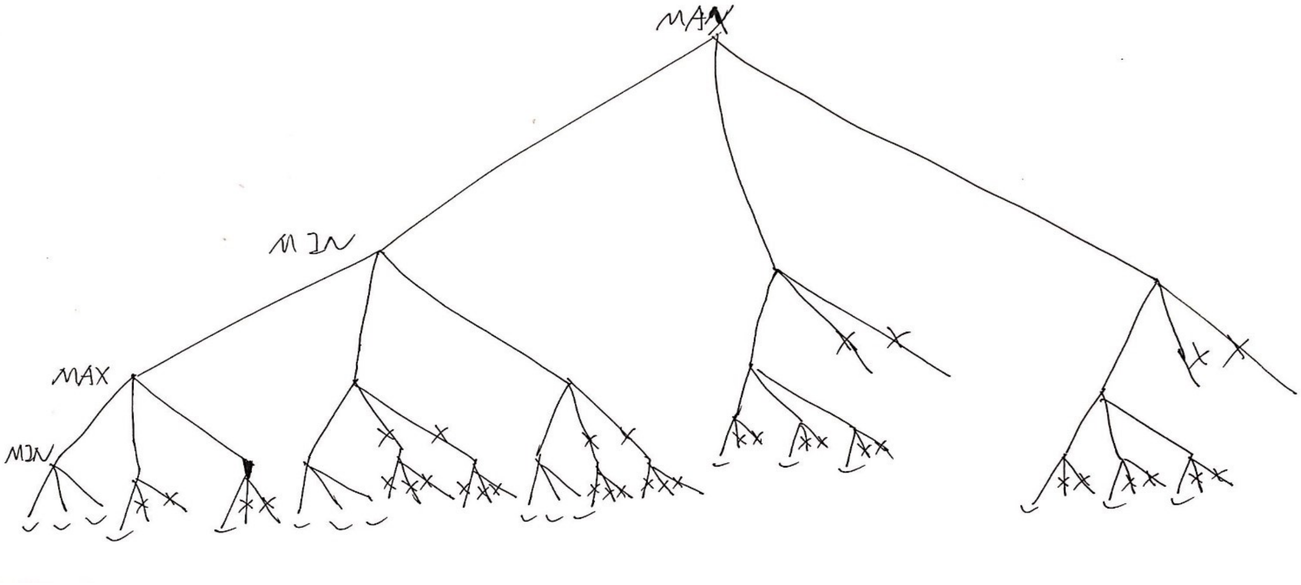
(a): the leaves of this game tree with all of the values from 0 to 15:



(b):

According to (a), there are 7 game tree leaves that are evaluated and 9 leaves that are pruned.

(c):



According to this game tree, there are 17 game tree leaves that are evaluated.

(d):

Time complexity for the best case is O(b^(d/2)) where b = branching factor and d = depth of the tree, and the base case is when all the preferred nodes are expanded first. The number of leaf node positions evaluated is about O(b*1*b*1*...*b) for odd depth and O(b*1*b*1*...*1) for even depth.

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