/*HaoweiLou, z5258575, assignment1-part2*/

Part2

Q1

a)

	start10	start12	start20	start30	start40
UCS	2565	Mem	Mem	Mem	Mem
IDS	2407	13812	5297410	Time	Time
A	33	26	915	Mem	Mem
IDA	29	21	952	17297	112571

b)

The Uniform Cost Search with Dijkstra's Algorithm has least efficiency in memory usage, since it does not work when the minimum number of moves is larger than 12, it also takes the most time for computation.

The Iterative Deepening Search is more efficient in memory usage than the Uninformed Cost Search, since it can solve the problem up to start20 without runs out of memory. But it is not efficient in time since it takes too much time to solve start30 and start 40.

The A star search is much more efficient in memory usage compare to IDS and UCS algorithm and consume less time. But it still cannot solve start30 and start40 because of memory usage.

The Iterative Deepening A* Search is the most efficient algorithm in these four, it can solve every problem without using all memories and need least time than others.

In overall, Iterative method is helpful in reducing memory usages and A start search will increase the time efficiency.

Q2

a)

	start50		start60		start64	
IDA*	50	14642512	60	321252368	64	1209086782
1.2						
1.4						
1.6						
Greedy	164	5447	166	1617	184	2174

b)

```
depthlim(Path, Node, G, F_limit, Sol, G2)  :-
    nb_getval(counter, N),
    N1 is N + 1,
    nb_setval(counter, N1),
    % write(Node),nl,   % print nodes as they are expanded
    s(Node, Node1, C),
    not(member(Node1, Path)),      % Prevent a cycle
    G1 is G + C,
    h(Node1, H1),
    W is 1.2,
    F1 is (2-W)*G1 + W*H1,
    F1 =< F_limit,
    depthlim([Node|Path], Node1, G1, F_limit, Sol, G2).
```

c)

	start50		start60		start64	
IDA*	50	14642512	60	321252368	64	1209086782
1.2	52	191438	62	230861	66	431033
1.4	66	116174	82	3673	94	188917
1.6	100	34647	148	55626	162	235852
Greedy	164	5447	166	1617	184	2174

d)

From the table, it shows that the IDA method takes least length of path from node to goal, but the greatest number of nodes expanded(G). hence it takes the greatest amount of time. As the value of w increase, the length of path also increases(N), but in the meanwhile, the number of nodes expanded(G) decrease.

Hence it results the time taken decreases and it can be said that the speed of algorithm increases as the value of w increase, because w increases leads the algorithm become similar to greedy algorithm, but it takes a longer length of path to find the goal. The smaller w leads algorithm more likely to be Iterative Deepening Search, hence takes more time with low speed but fewer length of path.