COMP 9414 Assignmet 2

Name: Jiaquan Xu unswid: z5174903

Q1

(a)

	START10	START12	START20	START30	START40
UCS	2625	MEM	MEM	MEM	MEM
IDS	2407	13812	5297410	TIME	TIME
A*	33	26	915	MEM	MEM
IDA*	29	21	952	17297	112571

(b)

- UCS will keep every nodes in memory and easy to run out of memory
- IDS will find an optimal solution, and use linear space, but it would take a little bit longer.
- A* will caculate the estimated total cost of cheapest solution through n, it will keep every nodes in memory, but it present well in time efficiency
- IDA* present well in both time and memory efficiency.

Q2

(a)(c)

	START 50		START 60		START 64	
IDA*	50	14642512	60	321252368	64	1209086782
1.2	52	191438	62	230861	66	431033
1.4	66	116342	82	4432	94	190278
1.6	100	34647	148	55626	162	235852
Greedy	144	5447	166	1617	184	2174

(b)

Changed code:

```
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),
   F1 is 0.8 * G1 + 1.2 * H1,
   F1 =< F_limit,
depthlim([Node|Path], Node1, G1, F_limit, Sol, G2).
```

(d)

The principle of greedy algorithm is always try to get closer to the goal (choose the smallest h(n)). While IDA* consider both cost from start to node (g(n)) and estimated cost from node to goal (h(n)). Use IDA* will find the optimal path while it takes longer to caculate. While use greedy search could get an answer quickly, but it could be stuck in loops.

Q3

(a)

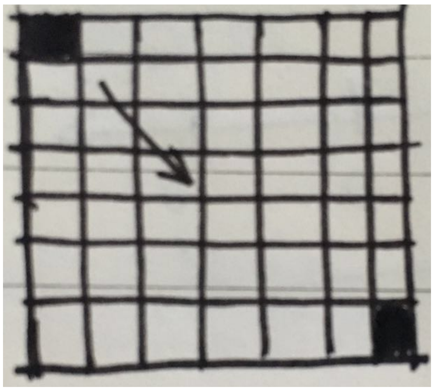
h(x, y, x_G, y_G) = $\sqrt{(x-xG)^2}$ + $\sqrt{(y-yG)^2}$

(b)

No, in the picture below, the black block takes 6 steps to the goal,

While $h_{LSD}(x,y,x_G,y_G) = 6\sqrt{2} > 6$

So h_{LSD} is not admissible any more.



(c)

No, still use the example in part(b), the black block takes 6 steps to the goal, while h(x, y, x_G, y_G) = 12 > 6

So h(x, y, x_G, y_G) in part(a) is not admissible.

(d)

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

Q4

(a)

n	Optimal sequence	M(n, 0)
1	[+ -]	2
2	[+ o -]	3
3	[+ o o -]	4
4	[+ + - -]	4
5	[+ + - o -]	5
6	[+ + o - -]	5
7	[+ + o - o -]	6
8	[+ + o o - -]	6
9	[+ + + - - -]	6
10	[+ + + - - o -]	7
11	[+ + + - o - -]	7
12	[+ + + o - - -]	7
13	[+ + + o - - o -]	8
14	[+ + + o - o - -]	8
15	[+ + + o o - - -]	8
16	[+ + + + - - - -]	8
17	[+ + + + - - - o -]	9
18	[+ + + + - - o - -]	9
19	[+ + + + - o - - -]	9
20	[+ + + + o - - - -]	9
21	[+ + + + o - - - o -]	10

(b)

Base case ($n = s^2$, $s \in Z$)

Each item represent the moving distance during each time steps.

The number of items represent number of time steps, that is M.

$1 + 2 + 3 + \ldots + s + (s-1) + (s-2) + \ldots + 1 + 0 = s^2$

'+' * s   '–' * s

Let $s^2 = n$, so $s = \sqrt{n}$, and $M(n, 0) = 2\sqrt{n} = 2\sqrt{n}$

if $s(s+1) < n <= (s+1)^2$, $s \in Z$:

we change from the base case formula and avoid breaking the continuity.

formula	n
$1 + 2 + 3 + \ldots + (s+1) + s + (s-1) + (s-2) + \ldots + 1 + 0 = s^2$	$(s+1)^2$
-1	$(s+1)^2 -1$
-1 -1	$(s+1)^2 -2$
-1 -1 -1	$(s+1)^2 -3$
…	…
$\underbrace{-1 \quad -1 \quad -1 \quad -1 \quad -1 \quad \ldots \quad -1}_{-1 * s}$	$(s+1)^2 -s$

That is $s(s+1) +1$

The number of items is still 2(s+1), means M(n,0) = 2s+2

if $s^2 < n <= s(s+1)$, $s \in Z$:

formula	n
$1 + 2 + 3 + \ldots + (s+1) + s + (s-1) + (s-2) + \ldots + 1 + 0 = s^2$	$(s+1)^2$
-(s+1)	$(s+1)^2 – (s+1)$
-(s+1) -1	$(s+1)^2 – (s+2)$
-(s+1) -1 -1	$(s+1)^2 – (s+3)$
…	…
-(s+1) -1 -1 -1 -1 … -1	$(s+1)^2 – (s+s)$

That is $s^2 +1$

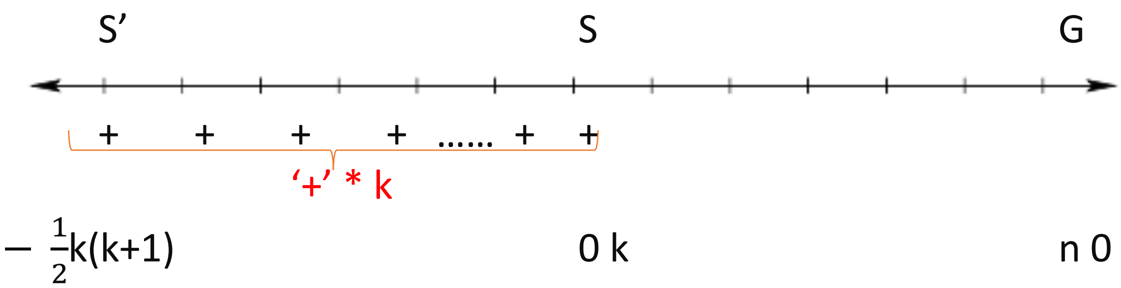
Remove this item !

The number of items is 2s+1, means M(n,0) = 2s+1

We conclude that:

$$M(n, 0) = \lceil 2\sqrt{n} \rceil = \begin{cases} 2s+1, & \text{if } \quad s^2 \quad < n \leq s(s+1) \\ 2s+2, & \text{if } s(s+1) < n \leq \ (s+1)^2 \end{cases}$$

(c)



We assume that the object start from S' with a velocity of 0, and keep speeding up to S,when it arrived S,it has the speed of k, the location of S' should be (1 + 2 + 3 +…+ k) = $-\frac{1}{2}$k(k+1) far away from S.

Let n' = S' to G, starts with 0 speed and ends with 0 speed.

M(n', 0) = M(n, k) + k

We use the formula from part(b):

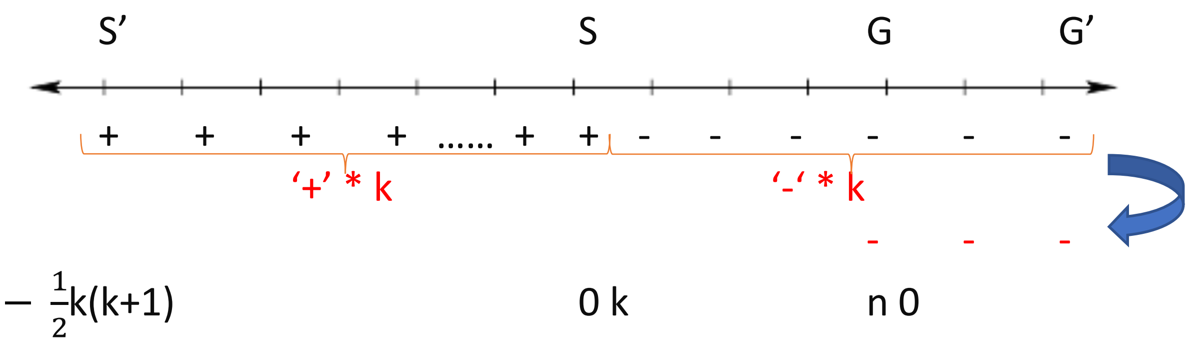
$$M(n', 0) = \lceil 2\sqrt{n'} \rceil$$

We already know that n' = n + $\frac{1}{2}$k(k+1)

So M(n, k) = M(n', 0) – k

$$= \left\lceil 2\sqrt{n + \frac{1}{2}k(k+1)} \right\rceil - k$$

(d)



If the distance between S and G is smaller than S to G'(the smallest distance for decelerating from volecity of k)= $\frac{1}{2}$k(k-1):

The object will first arrived at G' and turn back to G.

Let S'S = $\frac{1}{2}$k(k+1); SG = n; SG' = $\frac{1}{2}$k(k-1);

So GG' = SG' – SG = $\frac{1}{2}$k(k-1) – n

M(SG') = k

M(GG', 0) = $\left\lceil 2\sqrt{\frac{1}{2}\mathrm{k}(\mathrm{k}-1)-\mathrm{n}} \right\rceil$

Because the object will first arrived at G' and turn back to G.

The real time steps is M(SG', k) +M(GG', 0)

That is M(n, k) = k + $\left\lceil 2\sqrt{\frac{1}{2}\mathrm{k}(\mathrm{k}-1)-\mathrm{n}} \right\rceil$

(e)

h (r, c, u, v, rG, cG) = max(M(r-r_G, u), M(c-c_G, v))