COMP9414 Assignment 2

Fengting YANG

Z5089358

Question 1

a. The table below shows the result of running four algorithms.

G,N		***START10***	***START12***	***START20***	***START30***	***START40***
UCS	***G***	10	MEM	MEM	MEM	MEM
	N	2565	MEM	MEM	MEM	MEM
IDS	***G***	10	12	20	TIME	TIME
	N	2407	13812	5297410	TIME	TIME
A*	***G***	10	12	20	MEM	MEM
	N	33	26	915	MEM	MEM
IDA*	***G***	10	12	20	30	40
	N	29	21	952	17297	112571

b. While running the codes with these four algorithms, it is observed that A* and IDA* algorithms run faster than UCS which is faster than IDS. However, it looks that UCS uses the most memory as it can only run out the result of start10, A* also uses plenty of memory, but less than UCS. Memory usages of IDS and IDA* are hard to compare, but both are less than A*. IDA* runs out all the tests, so it is the most efficient algorithm.

Question 2

a. The table below shows the result of running IDA*, heuristic and greedy algorithms.

G,N	***START50***		***START60***		***START64***	
	G	***N***	***G***	***N***	***G***	***N***
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	33504	148	55626	162	235848
Greedy	164	5447	166	1617	184	2174

b. In the line

```
F1 is 0.4 * G1 + 1.6 * H1
```

Change it to

```
F1 is (2 – w) * G1 + w * H1
```

For example, when `w = 1.2`, the line should be

```
F1 is 0.8 * G1 + 1.2 * H1
```

c. See the table in question a

d. IDA* is the situation where w = 1. Greedy is the situation where w = 2. As w increases, N decreases, which means the algorithms become faster in speed since the quantity of states decreases. Only IDA* get the optimal solution. Conclusion in week 4 tutorial explain that. (It is guaranteed to be optimal when $0 \leqslant w \leqslant 1$ since it is equivalent to A*.

Question 3

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

b. (i). No. If we set $\Delta x = |x - x_G|$ and $\Delta y = |y - y_G|$, the total cost should be

$$max(\Delta x, \Delta y)$$

But the SLD is $\sqrt{\Delta x^2 + \Delta y^2}$.

It is easy to show that $\sqrt{\Delta x^2 + \Delta y^2} \geq max(\Delta x, \Delta y)$.

The equation holds iff $\Delta x = 0$ or $\Delta y = 0$.

That means, in general situation, SLD is bigger than the actual cost. So, it is not admissible.

(ii). No. From (i), it is also obvious that $\Delta x + \Delta y \geq max(\Delta x, \Delta y)$.

The equation holds iff $\Delta x = 0$ or $\Delta y = 0$.

That means, in general situation, Manhattan distance is bigger than the actual cost. So, it is not admissible.

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

Question 4

a. See the table below.

n	*Optimal seq*	*M(n,0)*
1	+-	2
2	+∘-	3
3	+∘∘-	4
4	++--	4
5	++-∘-	5
6	++∘--	5
7	++∘-∘-	6
8	++∘∘--	6
9	+++---	6
10	+++--∘-	7
11	+++-∘--	7
12	+++∘---	7
13	+++∘—∘-	8
14	+++∘-∘--	8
15	+++∘∘---	8
16	++++----	8
17	++++---∘-	9
18	++++--∘--	9
19	++++-∘---	9
20	++++∘----	9
21	++++∘---∘-	10

b. Since

$$\lceil 2\sqrt{n} \rceil = \begin{cases} 2s+1\,, if & s^2 < n \le s(s+1) \\ 2s+2\,, if & s(s+1) < n \le (s+1)^2 \end{cases}$$

Where s is the maximum speed.

When there is one rest, the n should hold the range in $s^2 < n \le s(s+1)$. When there are two rests, the n should hold the range in $s(s+1) < n \le s(s+2)$.

c. If starting with k at S and we set x to be the distance of acceleration, M(X,0) should be

$$M(X,0) = \lceil 2\sqrt{x} \rceil - k$$

Also, X should be $\frac{k(k+1)}{2}$

If we put these into the initial formula, it will become

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

d. If $n < \frac{k(k-1)}{2}$, it means we moves further than the goal. It needs to go reverse.

In this situation, M(n,k) = total time + reverse time – acceleration time

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

Simply, it will become

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

e. $h(r,c,u,v,r_G,c_G) = max(M(r_G - r, u), M(c_G - c, v))$