# Assignment 2

Name: Yu Feng ID:z5094935

**Qusetion1：**

**(a)**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Position | Start1 | Start2 | Start3 | Start4 | Start5 |
| N | 41 | 173 | 2409 | 9508 | 58419 |

Breadth first search, exclude repeated states:

**(b)**Breadth first search, include repeated states:

|  |  |  |
| --- | --- | --- |
| Position | Start1(N=) | Start2 |
| N | 139 | 1857 |

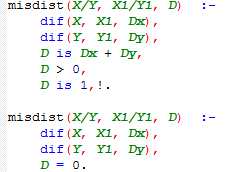
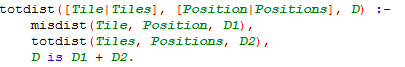
**(c)**

A\* with total Manhattan distance heuristic:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Position | Start1 | Start2 | Start3 | Start4 | Start5 |
| N | 6 | 11 | 19 | 61 | 213 |

**(d)**   
A\* with Misplace Tiles heuristic:

Changed parts:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Position | Start1 | Start2 | Start3 | Start4 | Start5 |
| N | 20 | 34 | 768 | Run out of time | Run out of time |

**(e)**   
Base on this result, the order from slowest to fastest is **c>a>b>d.**

**(f)**

**Position:**

|  |  |  |
| --- | --- | --- |
| 5 | 7 | 4 |
| 8 | 2 |  |
| 6 | 3 | 1 |

**Requiring 20: Requiring 23: Requiring 26:**

|  |  |  |
| --- | --- | --- |
| 5 | 7 | 4 |
| 8 | 2 | 1 |
|  | 6 | 3 |

|  |  |  |
| --- | --- | --- |
|  | 5 | 7 |
| 8 | 2 | 4 |
| 6 | 3 | 1 |

With A\* Manhattan Distance heuristic

|  |  |  |  |
| --- | --- | --- | --- |
| Requiring steps | **20** | **23** | **26** |
| N | 153 | 336 | 1305 |

**Qusetion2：**

**(a)**

**(b)**

1. **No**, the Straight-Line-Distance heuristic can be used for any direction. But diagonally move can just move by 45o. and even the degree is 45, the distance is not same, Straight-Line-Distance heuristic is , the other is 1.
2. **No**, because the agent can move diagonally. For example, if the agent moves from (0,0) to (1,1), for (a) heuristic, it cost 2 steps. While in (b) heuristic, it cost just one step.

**Qusetion3：**

|  |  |  |
| --- | --- | --- |
| n | time | sequence |
| 11 | 7 | [+ + + - o - -] |
| 12 | 7 | [+ + + o - - -] |
| 13 | 8 | [+ + + o - - o -] |
| 14 | 8 | [+ + + o - o - -] |
| 15 | 8 | [+ + + o o - - -] |
| 16 | 8 | [+ + + + - - - -] |
| 17 | 9 | [+ + + + - - - o -] |
| 18 | 9 | [+ + + + - - o - -] |
| 19 | 9 | [+ + + + - o - - -] |
| 20 | 9 | [+ + + + o - - - -] |
| 21 | 10 | [+ + + + o - - - o -] |

**(a)**

|  |  |  |
| --- | --- | --- |
| N | Time | sequence |
| 1 | 2 | [+ -] |
| 2 | 3 | [+ o -] |
| 3 | 4 | [+ o o -] |
| 4 | 4 | [+ + - -] |
| 5 | 5 | [+ + - o -] |
| 6 | 5 | [+ + o - -] |
| 7 | 6 | [+ + o – o - ] |
| 8 | 6 | [+ + o o - -] |
| 9 | 6 | [+ + + - - -] |
| 10 | 7 | [+ + + - - o -] |

**(b)**

The fastest way to get to the final is accelerate at the highest speed and slow down.

s is distance, a is acceleration, t is time. For initial speed is 0.

Because in this question, final speed must be 0.

t1 is accelerate time, t2 is slow down time

because speed from 0 to 0. t1 = t2. Let t = t1 +t2:

M(n,0) is the time t. n is the distant s, and in this question . let them take place of s and t, the least time used is:

In this question, all the things are discrete. So M(n,0) must be integer so:

**(c)**

As same as (b), just change the first formula to:

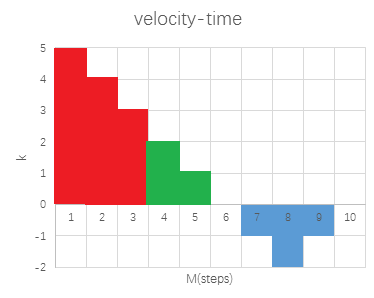
a = 1

Because it is discrete, the speed change need one step:

**(d)**

As the v-t diagram, the area of the blue part is the distance(n).

We can divide it into three parts. Red part is the n. Green part’s area equals blue part’s.

Let the red, green and blue M be t1, t2, t3 respectively.

Inferred:

**(e)**

Once a dimension gets the right position, let component of its velocity be 0, let another component get the goal. The result is the maximum of them.