(Q1)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | start10 | start12 | Start20 | Start30 | Start40 |
| UCS | 2565 | Mem | Mem | Mem | Mem |
| IDS | 2407 | 13812 | 5297410 | Time | Time |
| A\* | 33 | 26 | 915 | Mem | Mem |
| DA\* | 29 | 21 | 952 | 17297 | 112571 |

Discuss:

The ucsdijkstra algorithms has lowest efficiency, when it begins more than 10 it cost too much memory.

The ideepsearch algorithms has time complexity.

The A\* algorithms has good efficiency under 30, but when it starts more than 30 it costs too much memory .

The DA\* has best efficiency among all algorithm, which is the only one who can get the result of start40.

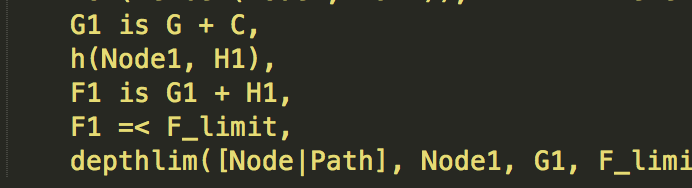
(Q2)

(a)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | start50 | | start60 | | start64 | |
| IDA\* | 50 | 14642512 | 60 | IDA\* | 50 | 14642512 |
| 1.2 | G=52 | N=191438 | G=62 | 1.2 | G=52 | N=191438 |
| 1.4 | G=66 | N=116342 | G=82 | 1.4 | G=66 | N=116342 |
| 1.6 | G=100 | N=33504 | G=148 | 1.6 | G=100 | N=33504 |
| Greedy | G=164 | N=5447 | G=166 | Greedy | G=164 | N=5447 |

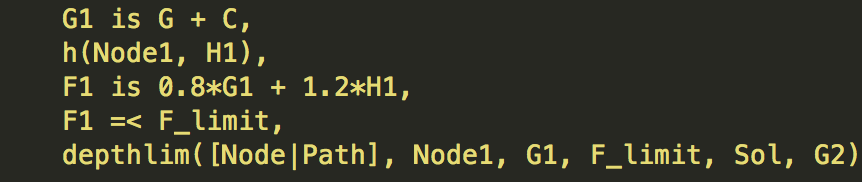
(b)

Initial code



change code

(F1=(2-w)\*G1+w\*H1)



(c)

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | start50 | | start60 | | start64 | |
| IDA\* | 50 | 14642512 | 60 | 321252368 | 64 | 1209086782 |
| 1.2 | G=52 | N=191438 | G=62 | N=230861 | G=66 | N=431033 |
| 1.4 | G=66 | N=116342 | G=82 | N=4432 | G=94 | N=190278 |
| 1.6 | G=100 | N=33504 | G=148 | N=55626 | G=162 | N=235848 |
| Greedy | G=164 | N=5447 | G=166 | N=1617 | G=184 | N=2174 |

(d)

When w is growing from 1.2 to 1.8, the number nodes are getting less while the path is growing. However, the speed of the process is getting faster while the quality is getting worse.

(Q3)

(a)

Another admissible heuristic:

h(x, y, xG, yG) =|x-xG|+|y-yG|

(b)(i)

No, the Straight-Line-Distance heuristic is not admissible. It is fact that straight line distance is always the shortest distance between 2 points. Considering the agent can only move step either up, down, left, right, and diagonally as well as the fact that diagonally move cost the same as move only vertically or horizontally, so the estimated path will be higher than the actual. For example, if a=4, b=4 then c=4 which is the value of the Straight-Line-Distance heuristic, however the actual value is 4 since every step of diagonally is and 4. So the estimated path is higher than the actual path, so it is not admissible.

a

b

c

(b)(ii)

No, my heuristic from part(a) is not admissible. According the question the agent in part(b) diagonal step is considered to have the same “cost” as a horizontal or vertical step, it is easy to find that diagonally move cost the same as move only vertically or horizontally, but my heuristic from part(a) is the sum of vertical and horizontal, so the estimated path is more than the actual path, which is contradict the definition of the admissible heuristic.

(b)(iii)

h(x, y, xG, yG) =max(|x-xG|,|y-yG|)

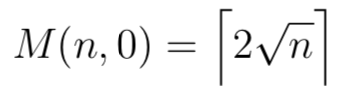
Q4

(a)

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

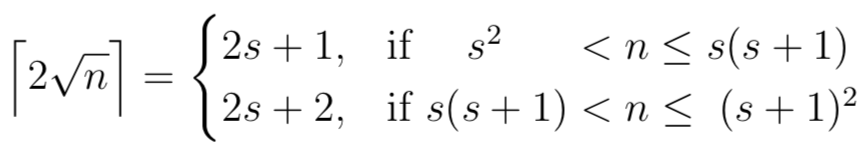
(b)

According to the formula for M(n,0) which is



(1)

and the identity formula



(2)

It is easy to find that the number of optimal sequence is equal and always match to the formula M(n,0).

The value of s is the maximum speed which is also represent the number of ‘+’. If n is perfect square like 1, 4, 9, and 16, s2 = n and M(n,0) = 2s. s2 is the least boundary and (s+1)2 is largest boundary of the formula, between the boundary there could be one or two, if n is smaller than s(s+1) then there would be 1 rest and M(n,0) = 2s+1, while n is larger than s(s+1) then there would be 2 rests and M(n,0) = 2s+2.

(c)

According to the question, we may start from S with velocity some k, if n>=(1/2)\*k(k-1), so we can stop or before the G since the shortest distance to accelerate the velocity from 0 to k is (1/2)\*k(k-1).

If M(n, 0) turn to M(n, k) so ,it won’t decrease it velocity which is the number of ‘-’, and it equal to the value of k, so

If x is the distance to decelerate the velocity from k to 0, then , and substituting this into (1) so

(d)

Since n<(1/2)\*k(k-1), it needs reverse some distance to get the velocity, otherwise it would over pass the goal G.

So:

(e)

Admissible heuristic for the original 2-dimensional GPGP game:

note: