Istanbul Technical University
Faculty of Computer and Informatics
Computer Engineering Department

BLG 435E
Artificial Intelligence
Homework I

Uğur Uysal - 150140012

November 9th, 2018

Contents

1 Quesiton -1

Q 1)

Public Transportation Recommendation System

- performance measure: time of recommended path, cost of recommended path etc.
- environment: digital, computer, roads.
- actuators: output screen.
- Sensors: input keyboard

environment: fully observable with cameras, etc.
stochastic.
sequential
dynamic

Goal based Agents. Because it interpret the changes of environment and optime a goal when it acts. Since our environment can be fully observable and stochastic and sequential our agent can come up with information about when traffic will change and recommend better paths.

2 Quesiton -2

Q2) Let heuristic $h(n)$ ant given that

$h(n) \leq c(n,a,n') + h(n')$, n is a state when n' is sucesor of n when action a is done (definition of consistency).

Let n is goal state then there wont be any action and succesor's so $h(n)=0$.

Let t is a node a optimal path to and n.

$h(t) \leq c(t, a, n) + h(n)$. (cost from $t \to n$, an optimal path with a action sequence)

Let add $g(t)$. to two sides.

$g(t) + h(t) \leq g(t) + c(t,a,n)$. -> cost to optimal path.

Let's say $+1$ $g(n)$

$h(t) \leq g(n) - g(t)$. } we can see that

$h(t) \leq$ cost from t to n. ($g(n)$ - total cost $g(t)$ - cost of comme to t)

we don't overestimate heuristic when ... consistent.

-> Show an admissible and non-consistent heuristic

Let's say we are searching on a path $n_0, n_1 \dots n_m$ (optimal solution length), ant $h(n_i) = \min(m-i, 1)$. it is a basics h is admissible, ($m-i$, actual cost and can not be greater then it).

Let $m = 2k$ ant i to k. (as evalue).

$h(n_i) = \min(k, k) \to k$

$h(n_{i+1}) = \min(2k-k-1, k+1) \to k-1$.

$h(n_i) \leq c(n, a, n') + h(n_{i+1})$.

$k \leq \text{cost} + (k-1)$

as we can see if cost is smaller then 1 (for this case) heuristic is not consistent.

3 Quesiton -3

3.1 Well Defined Form of Solo Test

In this homework, our objective is not same as original solo test game. We focus to find a state that there is not any available moves.

Initial State: is same as solo test.

Actions: Actions are movement of pegs and it's defined as jump over a neighbour(left,right,top,down) peg to an empty place.

State Space: State space is branches from initial state. And on path, from root to leaf nodes, of state-space graph, there is no possibility for state repetition.

Goal State: A state that there is no peg that can move.

Path Cost: In BFS and DFS costs are irrelevant but in A* cost is defined as number of pegs removed.

The Solution to the Problem : Check screen shot of BFS to see optimal(maximum number of pegs in the board) solution.

3.2 BFS

```
  OOO
  O.O
OOOOOOO
O.O.O..
OOOOOOO
  O.O
  O.O
BFS : Cost 6
BFS : Elapsed time: 0.045381
BFS : Number of nodes generated 70997
BFS : Number of nodes expanded 8113
BFS : Maximum number of nodes in frontier 62886
BFS : Number of pegs at final state 26
[ugur@ugur-pc hm1]$
```

Result of BFS.

3.3 DFS

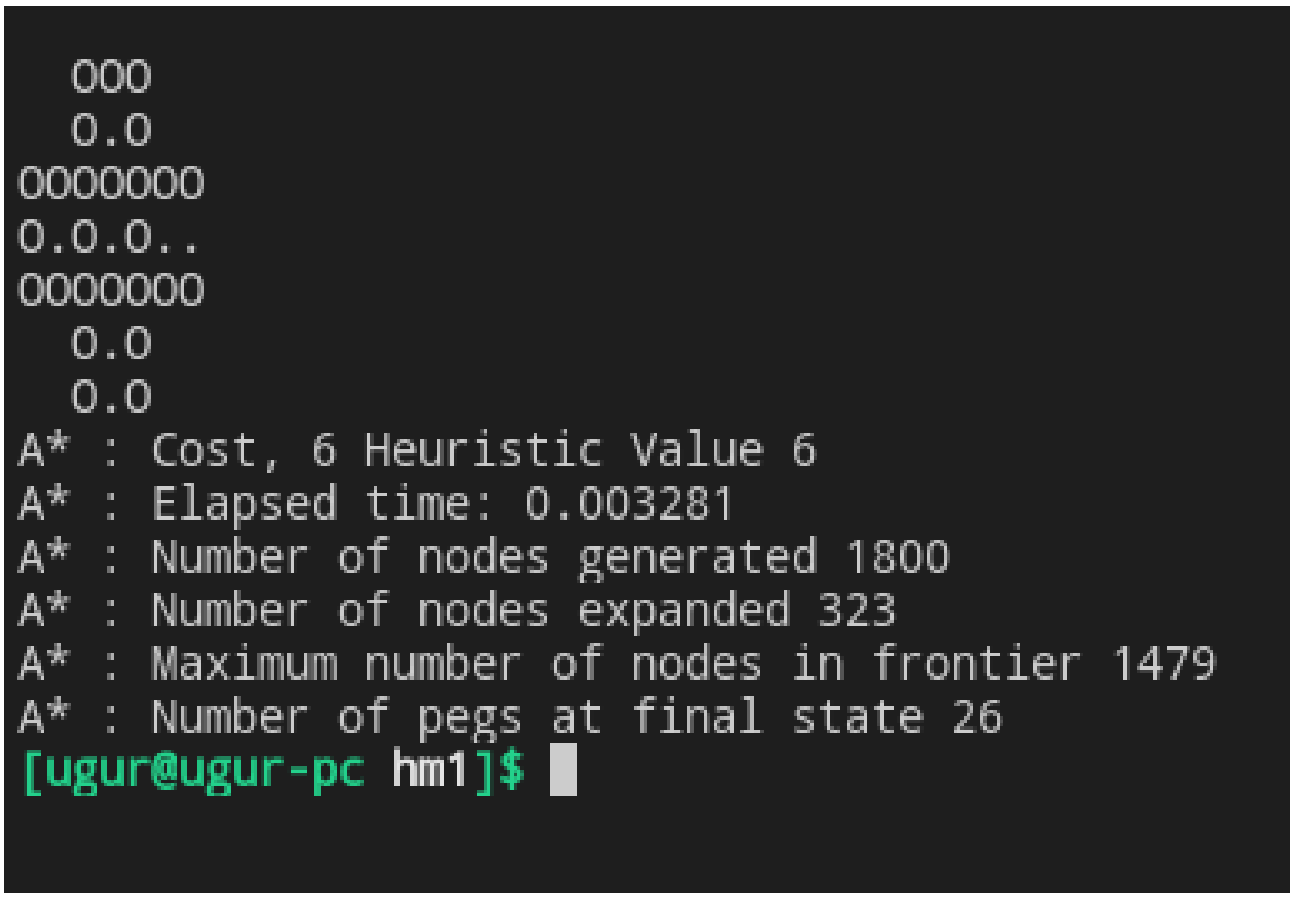
```
  OOO
  ...
.O...O
.......
O......
  ..O
  ...
DFS : Cost 25
DFS : Elapsed time: 0.000471
DFS : Number of nodes generated 133
DFS : Number of nodes expanded 26
DFS : Maximum number of nodes in frontier 109
DFS : Number of pegs at final state 7
[ugur@ugur-pc hm1]$
```

Result of DFS

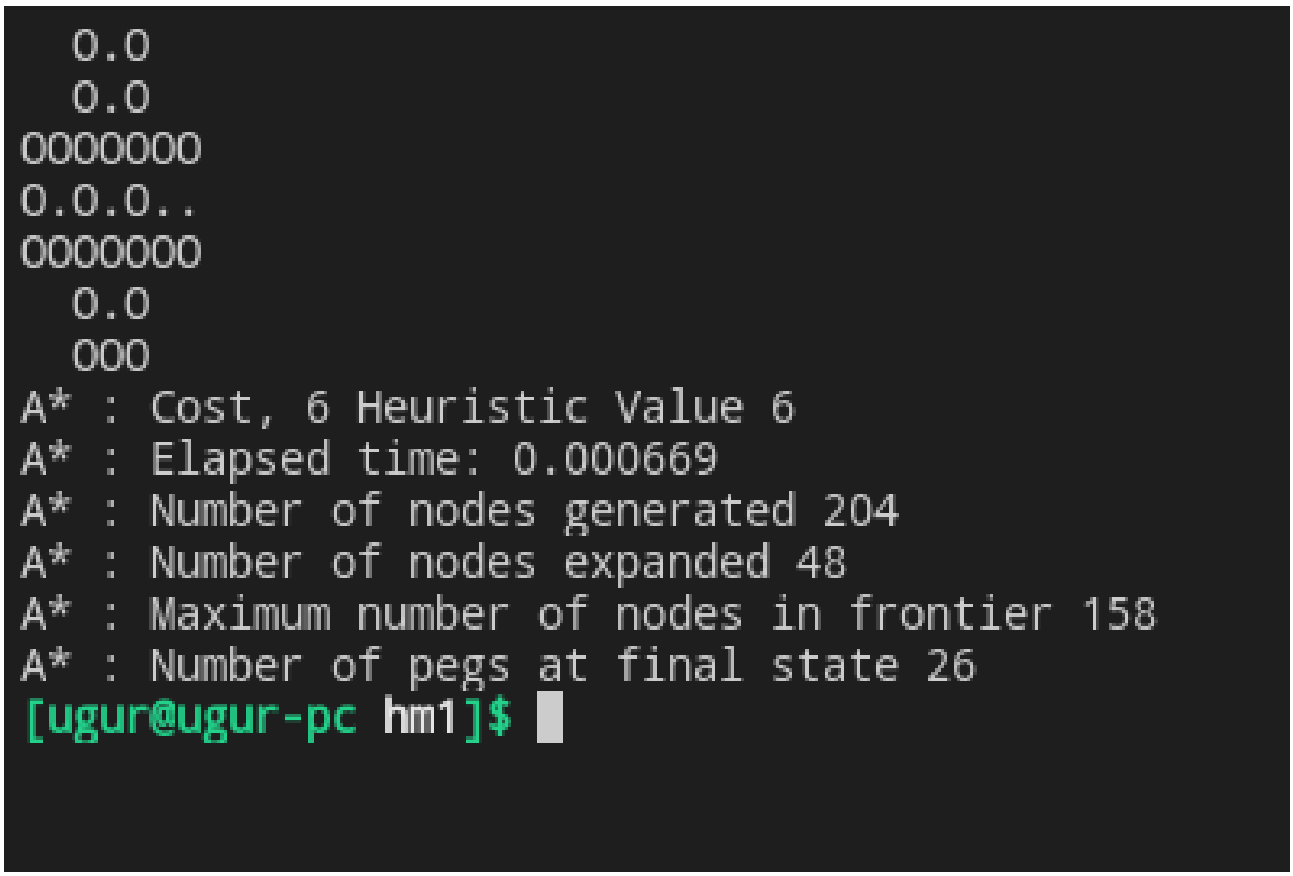
3.4 Analysis of BFS - DFS

As we can see from results, DFS is better than BFS in terms of memory and speed. Both algorithms are complete for this problem, because in a path at state-space graph there is no repeated states, so there is not any cycles. Otherwise DFS would not be complete. For this objective there is no issue with optimality. Both solutions are optimal, because they are solutions.

3.5 A*



Result of A* with heuristic I



Result of A* with heuristic II

3.6 Analysis of A*

As we can see from results, A* gives us optimal solution, actually same solution as BFS also more efficient in terms of speed and memory. That was expected result for A*. It gives better results with a heuristic that estimates cost near to best cost.

3.7 What if objective changes ?

If the objective changes towards to standard solo test game, BFS still would be able to find optimal solution in $O(b^d)$ where b is branching factor and d is depth of optimal solution) time complexity and $O(b^d)$ space complexity. DFS would be better than BFS in terms of memory($O(bd)$ space complexity same as time complexity of BFS) also it would be optimal solution unlike the other objective in the homework. A* with a good heuristic would be very efficient. It would outperform BFS in terms of time and space complexity. A* time complexity would depend on strength of heuristic. (Its complexity is relevant to error of heuristic)

4 How to run code

There is a single source code(150140012_hw_1.cpp) and makefile provided.

Code 1: How to compile and run source code

```
make
./150140012_hw_1 DFS
./150140012_hw_1 BFS
./150140012_hw_1 Astar h1
./150140012_hw_1 Astar h2
```