CSYE 7374

Autonomous Learning in Games

Assignment 2 – Classical Game AI

Professor: Nik Bear Brown

Due: February 15, 2020

TAs

Dikshant Rathi <rathi.d@husky.neu.edu>

Q1 (10 Points) Give a brief definition for the following:

1. Connected Component
2. Strong Connectivity
3. A\* Search (include Pseudocode)
4. Heuristic
5. Game Tree (include example)
6. Constraint satisfaction problem (CSP)
7. Dynamic Programming
8. Breadth First Search (BFS) (include Pseudocode)
9. Minimax algorithm (include example)
10. Alpha–beta pruning (include example)

Q2 (20 Points)

Consider the state space search problem shown in the graph. A is the start state and the shaded states are goals. Arrows encode possible state transitions, and numbers by the arrows represent action costs.

Note that state transitions are directed; for example, A -> B is a valid transition, but B -> A is not.

A picture containing drawing, necklace

Description automatically generated

Numbers shown in diamonds are heuristic values that estimate the optimal (minimal) cost from that node to a goal. For each of the following search algorithms, write down the nodes that are removed from fringe in the course of the search, as well as the final path returned. Because the original problem graph is a tree, the tree and graph versions of these algorithms will do the same thing, and you can use either version of the algorithms to compute your answer.

Assume that the data structure implementations and successor state orderings are all such that ties are broken alphabetically. For example, a partial plan S -> X -> A would be expanded before S -> X -> B; similarly, S -> A -> Z would be expanded before S -> B -> A.

(a) [4 pts] Depth-First Search

Nodes removed from fringe:

Path returned:

(b) [4 pts] Breadth-First Search

Nodes removed from fringe:

Path returned:

(c) [4 pts] Uniform-Cost Search

Nodes removed from fringe:

Path returned:

(d) [4 pts] Greedy Search

Nodes removed from fringe:

Path returned:

(e) [4 pts] A\* Search

Nodes removed from fringe:

Path returned:

Q3 (20 Points)

CSPs: Apple's New Campus

Apple's new circular campus is nearing completion. Unfortunately, the chief architect on the project was using Google Maps to store the location of each individual department, and after upgrading to iOS 6, all the plans for the new campus were lost!

The following is an approximate map of the campus:

A picture containing clock

Description automatically generated

The campus has six Offices, labeled 1 through 6, and six departments:

\_ Legal (L)

\_ Maps Team (M)

\_ Prototyping (P)

\_ Engineering (E)

\_ Tim Cook's office (T)

\_ Secret Storage (S)

Offices can be next to one another, if they share a wall (for an instance, Offices 1-6). Offices can also be across from one another (specifically, Offices 1-4, 2-5, 3-6). The Electrical Grid is connected to Offices 1 and 6. The Lake is visible from Offices 3 and 4. There are two \halves" of the campus South (Offices 1-3) and North (Offices 4-6).

The constraints are as follows:

i. (L)egal wants a view of the lake to look for prior art examples.

ii. (T)im Cook's office must not be across from (M)aps.

iii. (P)rototyping must have an electrical connection.

iv. (S)ecret Storage must be next to (E)ngineering.

v. (E)ngineering must be across from (T)im Cook's office.

vi. (P)rototyping and (L)egal cannot be next to one another.

vii. (P)rototyping and (E)ngineering must be on opposite sides of the campus (if one is on the North side, the other must be on the South side).

viii. No two departments may occupy the same office.

Constraints. Note: There are multiple ways to model constraint viii. In your answers below, assume

constraint viii is modeled as multiple pairwise constraints, not a large n-ary constraint.

(a) [4 pts] Which constraints are unary?

(b) [4 pts] In the constraint graph for this CSP, how many edges are there?

(c) [4 pts] Write out the explicit form of constraint iii.

(d) [4 pts] When enforcing arc consistency in a CSP, does the set of values which remain when the algorithm terminates depend on the order in which arcs are processed from the queue.

(e) [4 pts] In a general CSP with n variables, each taking d possible values, what is the maximum number of times a backtracking search algorithm might have to backtrack (i.e. the number of the times it generates an assignment, partial or complete, that violates the constraints) before finding a solution or concluding that none exists?

Q4 (10 Points)

For each of the game-trees shown below, state for which values of x the dashed branch with the scissors will be pruned. If the pruning will not happen for any value of x write \none". If pruning will happen for all values of x write “all".

We are assuming that nodes are evaluated left to right and ties are broken in favor of the latter nodes. A different evaluation order would lead to different interval bounds, while a different tie breaking strategies could lead to strict inequalities (> instead of >=). Successor enumeration order and tie breaking rules typically impact the efficiency of alpha-beta pruning.

A picture containing clock

Description automatically generated

(a) [2 pts] Explain why the triangles point up and down in the tree and why values less than or equal to one can be pruned.

(b-e) [2 pts each]

A close up of text on a white background

Description automatically generated

Q5 (10 Points)

A close up of a clock

Description automatically generated

Use the Bellman-Ford algorithm to find the shortest path from node A to all other nodes in the weighted directed graph above. *Show your work.*

Q6 (10 Points)

Use Kruskal's algorithm to find a minimum spanning tree for the connected weighted graph below:

A close up of a clock

Description automatically generated

What is the Time Complexity of Kruskal's algorithm?

Q7 (10 Points)

Use Prim's algorithm to find a minimum spanning tree for the connected weighted graph below*. Show your work.*

A close up of a clock

Description automatically generated

What is the Time Complexity of Prim's algorithm?

Q8 (10 Points)

Find shortest path from A to F in the graph below using Dijkstra's algorithm. *Show your steps.*

A close up of a clock

Description automatically generated