CSE 571 Fall 2022

Homework 2

Due *September 8, Thursday* online

Homework Instructions: Read Carefully

1. Only typed answers will be accepted. Solutions with **ANY** written part (except for hand-drawn illustrative figures) will not be given any credits. If you need to write equations, use "Insert->Equation" with *Word*. *LaTeX (e.g., Overleaf)* also supports equation typesetting.

2. Do **NOT** include questions themselves in your answers. Failing to do some may result in failing the plagiarism check.

3. Answers without explanations will **NOT** be given any credits.

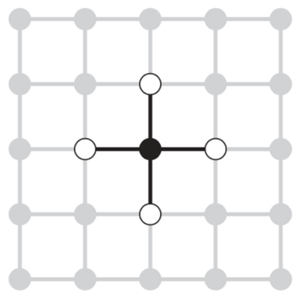
Be precise and concise in your answers. You may add hand-drawn figures when necessary.

Exercise 1.1 (6pt)

Prove each of the following statements, or give a counterexample:

a. Breadth-first search is a special case of uniform-cost search.
b. Depth-first search is a special case of best-first search.
c. Uniform-cost search is a special case of A$*$ search.

Exercise 1.2 (18pt)


from textbook

Consider the unbounded version of the regular 2D grid shown above. The start state is at the origin, *(0, 0)*, and the goal state is at *(x, y)*. The agent can choose action North, South, East, West ***or Stay*** in any state. Consider ***tree search*** below (unless noted otherwise):

a. What is the branching factor *b (i.e., maximum possible transitions from each state)*?
b. How many distinct states are there at depth *k* (for *k > 0*) in the search tree?

c. What is the maximum number of nodes expanded by breadth-first search?
d. What is the maximum number of distinct nodes expanded by breadth-first search?
e. What is the maximum number of nodes expanded by breadth-first ***graph*** search?
f. Is $h = |u - x| + |v - y|$ an admissible heuristic for a state at (u, v)? Explain.
g. How many nodes are expanded by A∗ graph search using *h in (f)*?
h. Does h remain admissible if some links are removed? Explain.
i. Does h remain admissible if some links are added between nonadjacent states? Explain.

Exercise 1.3 (12pt)

n vehicles occupy squares $(1, 1)$ through $(n, 1)$ (i.e., the bottom row) of an $n \times n$ grid. The vehicles must be moved to the top row but in reverse order; the vehicle i that starts in $(i, 1)$ must end up in $(n-i+1, n)$. On each time step, every one of the n vehicles can move one square up, down, left, or right, or stay put; but if a vehicle stays put, one other adjacent vehicle (but not more than one) can hop over it. Any vehicle is allowed to hop at most one other vehicle at a time. Two vehicles cannot occupy the same square.

a. (2pt) Calculate the size of the state space as a function of n.
b. (2pt) Calculate the branching factor as a function of n.
c. (2pt) Suppose that vehicle i is at (x_i, y_i); write a ***nontrivial*** admissible heuristic h_i for the number of moves it will require to get to its goal location $(n-i+1, n)$, ***assuming no other vehicles are on the grid.***
d. (6pt) Which of the following heuristics are admissible for the problem of moving all n vehicles to their destinations? ***Explain either way for each proposal below***.
(i) $\sum_i h_i$
(ii) $max\,\{h_1 \ldots h_n\}$
(iii) $min\,\{h_1 \ldots h_n\}$

Exercise 1.4 (6pt)

from wiki

Once upon a time a farmer went to a market and purchased a wolf, a goat, and a cabbage. On his way home, the farmer came to the bank of a river and rented a boat. But crossing the river by boat, the farmer could carry only himself and a single one of his purchases: the wolf, the goat, or the cabbage.

If left unattended together, the wolf would eat the goat, or the goat would eat the cabbage.

The farmer's challenge was to carry himself and his purchases to the far bank of the river, leaving each purchase intact. He also wants to make as fewer trips as possible.

a. Formulate the problem in a search formulation. **Ensure that only the relevant information is encoded in your state.** What is the size of the state space?
b. Design a non-trivial admissible heuristic.

Exercise 1.5 (8pt)

In our class, we have considered combining BFS and DFS to give iterative deepening (for tree search). Let us do the same with UCS and DFS to create cost-iterative deepening now. Write the pseudocode of cost-iterative deepening and analyze it properties (i.e., completeness, optimality, computational and space complexity)**. Use notions that are consistent with our lecture slides whenever possible.**