COMPSCI 250: Fall 2023
Homework 5

David A. Mix Barrington, Ghazaleh Parvini

Due Date: Monday, November 20;
Late Day: Tuesday, November 21

This assignment has 8 problems. There is also 1 Extra Credit problem. The extra credit is 10 points.

Please submit a single PDF file, with the problems in order (as below), and legible. Look at your PDF before submitting it – it is fine to scan or photograph a handwritten document but it the graders can't read it, they won't grade it.

Please assign pages to problems in Gradescope. Graders will click on the problem number. If no page shows up because it's not assigned, the assumption is you have not solved the problem.

Be sure you are doing Problems in the book and not Exercises: the numbers should start with P rather than E.

For full credit, show your work, explaining your reasoning. This also helps assign partial credit.

You are responsible for following the academic honesty guidelines on the Grading and Requirements page. This means that what you present must be your own work in presentation, and you must acknowledge all sources of aid other than course staff and the textbook. You will get 2 extra points if you typeset your Homework.

(12 points) **Problem 9.5.2**

Suppose that a directed acyclic graph has maximum path length d and that no node has more than b neighbors. What is the largest number of node visits that could occur in a depth-first search of this graph? Show an example of such a graph that has only $bd + 1$ nodes and has the maximum number of node visits.

(12 points) **Problem 9.6.8**

Let G be an undirected graph with n nodes that contains two nodes s and t, such that the shortest path from s to t has more than $n/2$ edges. Prove that there exists a node u, different from s and t, such that every path from s to t passes through u.

(10 points) **Problem 9.8.8**

Here is a single-step distance matrix for a weighted directed graph (shown in Figure 9-16) that indicates driving times among six locations in Massachusetts during Friday rush hour. The locations are Amherst (Amh), Hadley (Had), North Amherst (NAm), North Hadley (NHa), Northampton (Ntn), and Sunderland (Sun). An entry of – in the table indicates that there is no edge. (You may ignore the numbers next to each town name in Figure 9-16 – these will be used in Problem 9.9.7.)

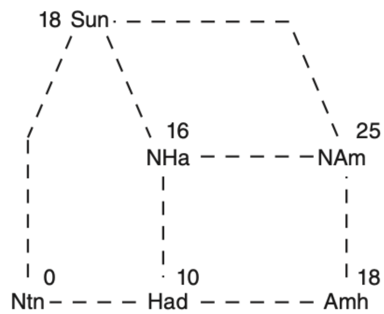

Figure 9-16: A directed graph representing some towns in Massachusetts.

	Amh	Had	NAm	NHa	Ntn	Sun
Amh	0	15	15	--	--	--
Had	12	0	--	8	30	--
NAm	19	--	0	10	--	14
NHa	--	10	9	0	--	12
Ntn	--	15	--	--	0	20
Sun	--	--	11	12	22	0

(10 points) **Problem 9.8.9**

Our uniform-cost search is very similar to **Dijkstra's Algorithm**, the best-known method of solving the single-source shortest path problem. In one formulation, Dijkstra's Algorithm keeps an array D indexed by the nodes of the graph, so that $D(x)$ indicates the shortest so far known distance from s to x. Initially, we set $D(s)$ to 0 and $D(x)$ to ∞ for all other nodes x. Another boolean array classifies each node as "explored" or "unexplored", with all nodes initially unexplored. A step of the algorithm is as follows:

- Find the unexplored node x with the smallest value of $D(x)$.
- For every edge out of x, to node y with cost $c(x, y)$, compute $D(x) + c(x, y)$.
- If any of these values are smaller than the corresponding $D(y)$, reset $D(y)$ to the new value.
- Mark x as explored.

The algorithm ends if a goal node is marked explored, or if $D(x)$ for every unexplored x is ∞.

(a) Explain how this algorithm corresponds to a uniform-cost search of the same graph, in particular how each step of one corresponds to a step of the other.

(b) Uniform-cost search uses a priority queue. Explain how a priority queue can be used to improve the running time of Dijkstra's Algorithm.

(12 points) **Problem 9.9.7**

Conduct an A^* search of the weighted directed graph of Problem 9.8.8, with start node North Amherst and goal node Northampton. Use the heuristic function h defined by $h(Ntn) = 0$, $h(Had) = 10$, $h(NHa) = 16$, $h(Amh) = 18$, $h(Sun) = 18$, and $h(NAm) = 25$. (This heuristic represents the time needed to drive to Northampton with no traffic.) Figure 9-16 shows the heuristic value for each node, and the driving times are given in a table in Problem 9.8.8.) Has the traffic in the original problem affected the optimal route from North Amherst to Northampton?

(12 points) **Problem 5.1.6**

Let A be any finite alphabet. The language $A^{\leq k}$ is defined to be the union $A^0 + A^1 + \ldots + A^k$, the set of all strings over A with length at most k. Prove by induction on all naturals k that the languages $(\emptyset^* + A)^k$ and $A^{\leq k}$ are equal.

(10 points) **Problem 5.2.4**

Construct a regular expression denoting the language of strings in $\{a, b\}^*$ that have a number of b's that is divisible by 3.

(10 points) **Problem 5.4.7**

Prove that for any two languages S and T, $(ST)^*S = S(TS)^*$. Use induction on the definition of the Kleene star languages.

Extra Credit

(10 points) **Problem 5.5.6**

If L is any language, we define its **substring language** $Sub(L)$ to be the set of all strings y such that y is a substring of any string $x \in L$. Prove that if S is any regular expression, then $Sub(L(S))$ is a regular language. Give a recursive algorithm to produce a regular expression for this language.