CSCI 470 Homework 5

Please write your solutions in the LaTeX. You may use online compiler such as Overleaf or any other compiler you are comfortable with to write your solutions in the LaTeX.

Due date: Monday Nov. 27, 2023, 9:10 AM EST.

Please submit a PDF (preferably written in LaTeX) or a scanned copy of your handwritten solutions to Homework 05 on Canvas. Points will be deducted if handwritten solutions are not legible. **Also, please bring a physical copy of your homework when the class meets on Nov 27**. Apparently, it's faster to annotate your submissions on paper compared to annotating PDFs on Canvas. This will help the TA to grade your submissions sooner.

Please note that the figures referenced in this homework are from the book, CLRS, 3rd edition.

You can use the LaTeX submission template I have shared along with the homework. There are two .tex files ("macros.tex", and "main.tex"). You can upload the zipped folder directly to Overleaf or create a blank project on Overleaf and upload macros.tex and main.tex files, and edit main.tex to write your solutions. "macros.tex" is mostly for macros (predefined commands).

Single-Source Shortest Paths [50 points]

Problem 5-1. (20 points) Run the Bellman-Ford algorithm on the directed graph of Figure 24.4 (pg. 652 CLRS), using vertex z as the source. In each pass, relax edges in the same order as in the figure, and show the d and π values after each pass. Now, change the weight of edge (z, x) to 4 and run the algorithm again, using s as the source.

Problem 5-2. (10 points) Run DAG-SHORTEST-PATHS on the directed graph of Figure 24.5, using vertex r as the source.

Problem 5-3. (20 points) Run Dijkstra's algorithm on the directed graph of Figure 24.2, first using vertex s as the source and then using vertex z as the source. In the style of Figure 24.6, show the d and π values and vertices in set S after each iteration of the **while** loop. You can list the value of d π in a table, instead of shading the edges to show the change in the predecessors of each node. Note that there will be $|V| = 5$ iterations, while listing the values of d and π in the table.

Dynamic Programming [50 points]

Problem 5-4. (15 points) Consider a modification of the rod-cutting problem in which, in addition to a price p_i for each rod, each cut incurs a fixed cost of c. The revenue associated with a solution is now the sum of the prices of the pieces minus the costs of making the cuts. Give a dynamic-programming algorithm to solve this modified problem.

Problem 5-5. (15 points) Determine an LCS of $\langle 1, 0, 0, 1, 0, 1, 0, 1 \rangle$ and $\langle 0, 1, 0, 1, 1, 0, 1, 1, 0 \rangle$.

Problem 5-6. (20 points) Give pseudocode to reconstruct an LCS from the completed c table and the original sequences $X = \langle x_1, x_2, ..., x_m \rangle$ and $Y = \langle y_1, y_2, ..., y_n \rangle$ in $O(m + n)$ time **without** using the b table.

Extra Credit [30 points]

Problem 5-7. (15 points) Give a memoized version of LCS-LENGTH that runs in $O(mn)$ time.

Problem 5-8. (15 points) Modify MEMOIZED-CUT-ROD to return not only the value but the actual solution, too.

Here is the MEMOIZED-CUT-ROD procedure for your reference.

```
MEMOIZED-CUT-ROD(p, n)
let r[0 . . n] be a new array
for i = 0 to n
    r[i] = −∞
return MEMOIZED-CUT-ROD-AUX(p, n, r)
```

```
MEMOIZED-CUT-ROD-AUX(p, n, r)
if r[n] ≥ 0
    return r[n]
if n == 0
    q = 0
else
    q = −∞
    for i = 1 to n
        q = max(q, p[i] + MEMOIZED-CUT-ROD-AUX(p, n − i, r))
r[n] = q
return q
```