# COMP 221 Homework 8

Due: Friday, November 22, by 11:59 pm

## 1 Guidelines

Please **type up** and submit a PDF of your solutions. If you have to draw any pictures for a question (such as trees, graphs), you can do so **neatly** and then add a picture of it to your PDF file.

**Programming Questions:** Please use Python or Java.

**Do your own work:** Homework should be individual work. You may discuss problems with other people, but you must write up your solutions yourself. I will not say that the web is off limits, but handing in solutions you find online as if they are your own is also not acceptable. Speaking of, most of these questions are shamelessly stolen from Susan :)

Make sure you have your name on your homework!

- **Question 1** Question about alphabet search (letters to numbers)

  Show the steps of the `ShortestAugmentingPath` algorithm for the Max-Flow problem, operating on the graph below. Assume that A is the source, and G is the sink.

  For each pass through the loop, tell me:

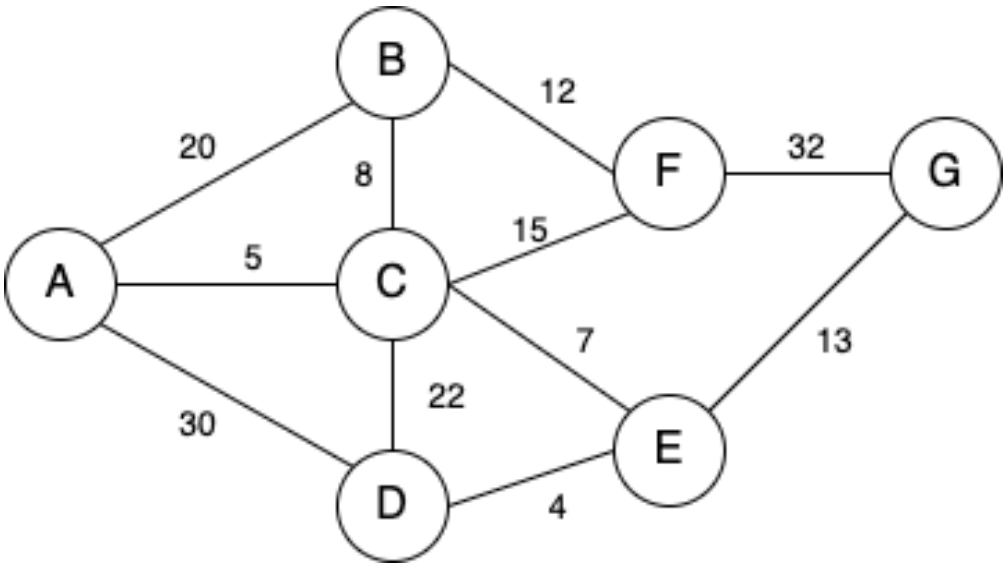
  (1) the path that the `SpecialBFS` Algorithm would find

  (2) the minimum capacity along that path

  (3) the total cumulative flow

  (4) a diagram or table showing the edge values for the graph and how they would be updated.

  For `SpecialBFS`, assume that neighbors are processed in *alphabetical order*.




- **Question 2 - Knapsack**

  Given the table below showing items, their corresponding values and their corresponding weights, and a knapsack **capacity of 8**, use the dynamic programming approach we discussed in class to find the set of items that would fit in the knapsack with the maximum value.

| Item | Value | Weight |
|---|---|---|
| 1 | 5 | 2 |
| 2 | 6 | 3 |
| 3 | 3 | 1 |
| 4 | 2 | 2 |
| 5 | 15 | 4 |
| 6 | 3 | 1 |

- **Question 3 - Programming**

  Implement the dynamic programming algorithm for finding the longest common subsequence (LCS) for two strings.

  Test your program on a variety of inputs, including these two randomly generated DNA sequences:

  s1: CAGGACGGCGTGCGCGTTCCATGTAAACCTGTCATAACTTACCTGAGACT

  s2: AGTTGGAAGTGTGGCTAGATCTTTGCTCAC