CSCI 411 - Advanced Algorithms and Complexity
Assignment 3

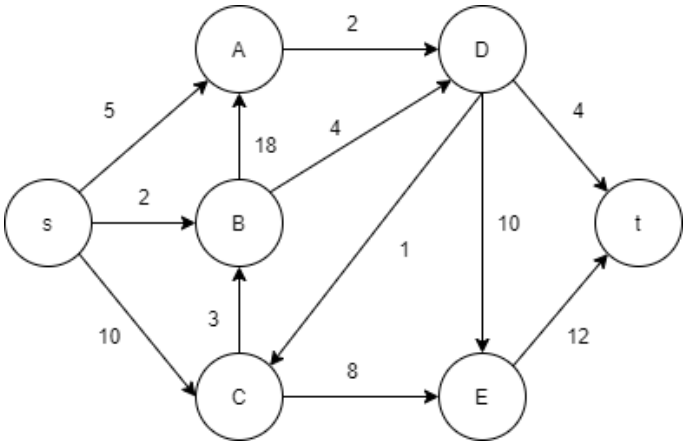
September 12, 2022

Solutions to the written portion of this assignment should be submitted via PDF to Blackboard. Make sure to justify your answers. C++ code should be submitted on both Blackboard and turnin. Both parts of the assignment are due before **September 18th at 11:59 pm**.

There will likely be time in class to discuss these problems in small groups and I highly encourage you to collaborate with one another outside of class. However, you must write up your own solutions **independently** of one another. Feel free to communicate via Discord and to post questions on the appropriate forum in Blackboard. Do not post solutions. Also, please include a list of the people you work with at the top of your submission.

Written Problems

1. Use the Edmonds-Karp algorithm to find a maximum flow in the network below. Edge capacities are given.



   (a) (7 pts) Draw the residual network associated with the flow you found.

   (b) (5 pts) What is the value of this flow?

   (c) (8 pts) Find a cut (S, T) in the network that shows the flow you found is maximum.

2. Given two sequences A and B, we would like to determine their longest common subsequence. Note that, while the elements of a substring must be consecutive, the elements of a subsequence do not.

   (a) (15 pts) Describe the optimal substructure of this problem. In particular, define the longest common subsequence of A and B in terms of the solution for shorter sequences. Justify your answer.

   (b) (10 pts) Write pseudocode for a function `findLCS(A, B)` which returns the length of the longest common subsequence of A and B.

   (c) (5 pts) Analyze the asymptotic run time of your algorithm.

3. Given a set of coin denominations C, determine the fewest number of coins required to produce an amount of money m.

   (a) (15 pts) Describe the optimal substructure of this problem. In particular, define the solution for an amount of money m in terms of solutions for amounts $\mu < m$. Justify your answer.

   (b) (10 pts) Write pseudocode for a function `makeChange(m, C)` which returns a list indicating the number of coins of each denomination required to reach the target value m using the fewest total number of coins.

   (c) (5 pts) Analyze the asymptotic run time of your algorithm in terms of both m, the target amount of money, and $|C|$, the number of available coin denominations.

Coding Problem

(20 pts) Write a C++ implementation of the pseudocode you developed for problem (3b) and submit to Blackboard and to turnin as assignment_3.cpp. You may find the skeleton code in assignment_3_skeleton.cpp on Blackboard helpful.

- Input will come from cin
  - The first line will contain two integers, $1 \leq n \leq 50$ and $3 \leq m \leq 15$, separated by a space.
  - n is the number of coin denominations and m is the number of target amounts to follow.
  - The next line contains n space separated integers representing coin denominations between 1 and 10000.
  - The next m lines each contain a single integer between 1 and 12345678.
- Print output to cout
  - For each target amount, if the amount can be achieved with the given coin denominations, print n space separated integers on a new line representing the number of each available coin used to generate change.
  - The total number of each coin should appear in the same order in which the denominations were given.
  - Print 0 when a particular denomination is not used.
  - If more than one way exists to make change using the same number of coins, print results for the approach using coins appearing sooner in the given order.

- If the target amount cannot be generated using the given denominations, print 0 for all denominations.
- There should be no trailing spaces.
- There should be a single newline at the end of the output.

Examples

Example 1:

Input:
4 3
1 5 10 25
11
42
1234567

Expected output:
1 0 1 0
2 1 1 1
2 1 1 49382

Example 2:

Input:
4 3
1 5 12 25
16
91
1234567

Expected output:
1 3 0 0
0 1 3 2
0 1 1 49382

Example 3:

Input:
4 3
2 4 6 8
9
10
12345678

Expected output:
0 0 0 0
1 0 0 1
0 0 1 1543209