COMPSCI 320SC 2024 Midterm Test

Put the answers in the space below the questions. Write clearly and *show all your work*!

Marks for each question are shown below and just before each answer area.

This 60 minute test is worth 10% of your final grade for the course.

Question #:	1	2	3	4	Total
Possible marks:	5	5	5	5	20
Awarded marks:					

1. (a) Write a formal definition for the big-Theta notation, $f(n) = \Theta(g(n))$. ***(2 marks)***

There exist a pair of constants $c_2 > c_1 > 0$ and constant $n_0 \geq 0$, such that for all $n > n_0$, we have $c_1 \cdot g(n) \leq f(n) \leq c_2 \cdot g(n)$.
Also acceptible to say that both $f(n) = O(g(n))$ and $g(n) = O(f(n))$ hold.

(b) Let p be a nonnegative function in $\Theta(n)$, let q be a nonnegative function in $O(n^3)$, let r be a nonnegative function in $\Omega(n^2)$, and let $s = pq + r$. In other words, $s(n) = p(n) \cdot q(n) + r(n)$. Give an example of a function that might be s, and of a function that cannot be s. What else can you say about function s? ***(3 marks)***

By the multiplication rule we have that pq is $O(n^4)$. We cannot use the addition rule between pq and r, but using the fact that pq is non-negative we an conclude that $pq + r \geq r$ and hence s is $\Omega(n^2)$. We cannot give any upper bound because r may be arbitrarily large. An example of a function that may be s is n^3, and an example of a function that cannot be s is n.

2. For the following questions, use the following definition of Integer Programming.

Definition: *Let A be an $m \times n$ matrix with integral coefficients and $b \in Z^m$ as input. Assuming column vectors, solving:*

$$\exists x\{x \in \{0, 1, 2, \dots\}^n \text{ and } Ax = b\}$$

is called an instance of the ***Integer Programming*** *problem.*

(a) With vectors $y = (1, -3, 4)$ and $z = (5, 1, 2)$ determine the dot product $y \cdot z$.

(***1 mark)***

$5 - 3 + 8 = 10$

(b) With $M = \begin{bmatrix} 1 & 0 \\ -2 & 3 \end{bmatrix}$ compute M^4 efficiently. (***1 mark)***

$$M^2 = \begin{bmatrix} 1 & 0 \\ -8 & 9 \end{bmatrix}$$

$$M^4 = (M^2)^2 = \begin{bmatrix} 1 & 0 \\ -80 & 81 \end{bmatrix}$$

(c) Compute the matrix times vector product $\begin{bmatrix} 5 & 3 & 0 \\ -1 & 2 & 1 \\ 7 & 0 & -2 \end{bmatrix} \begin{bmatrix} 4 \\ 1 \\ 0 \end{bmatrix}$. (***1 mark)***

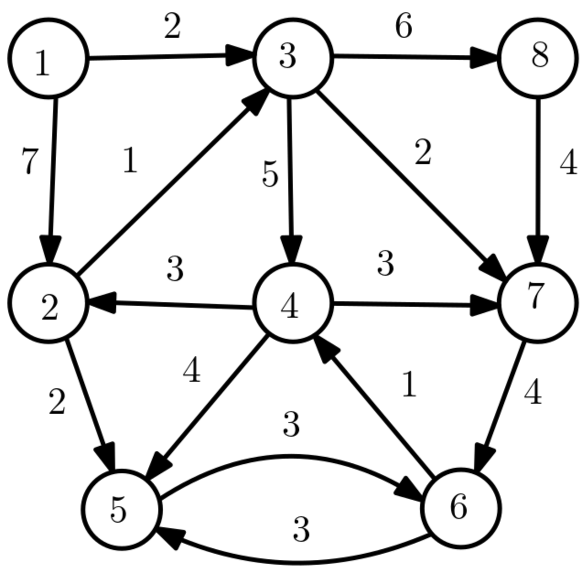
$$\begin{bmatrix} 23 \\ -2 \\ 28 \end{bmatrix}$$

(d) Let A be an adjacency matrix for a digraph G of order n and b a vector of n non-negative integers as input to an Integer Programming instance. If there is a solution, what do we know about G with respect to b? Briefly explain your answer in terms of vertex labels. (***2 marks)***

If there is a solution to the Integer Programming problem then we know that there is a labeling of the vertices (x) with non-negative integers such that the sum of the out-neighbors' labels equals the label of each vertex. That is, each row i of the matrix A times x selects those elements of x to sum to the i-th element of b.

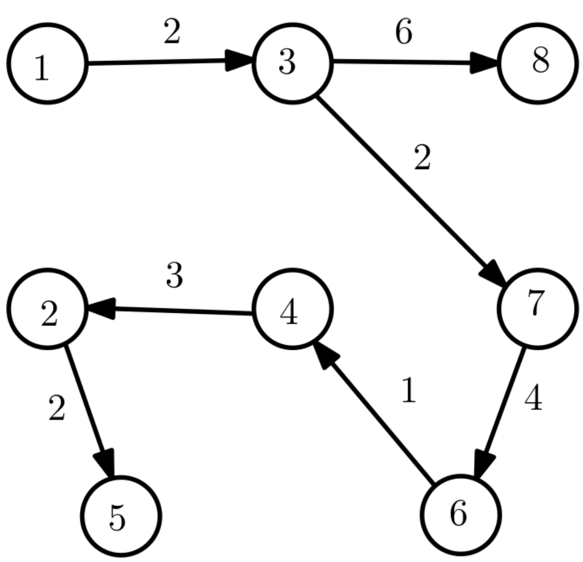
3. Apply greedy algorithms for the following two problems that were presented in lectures.

   (a) Compute a minimum weighted arborescence of the following digraph rooted at node 1.



(3 marks)

The minimum cost is 20:



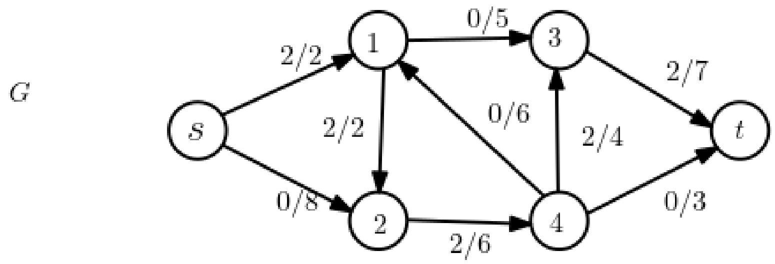
(b) Suppose we have a coded message with six symbols $(\alpha, \beta, \gamma, \delta, \epsilon, \zeta)$ of frequencies 0.40, 0.25, 0.10, 0.10, 0.10, and 0.05, respectively. We want to produce a Huffman code with the least average bits per letter (ABL). Compute an optimal prefix code and its ABL. ***(2 marks)***

$0.4 \cdot 1 + 0.25 \cdot 2 + (0.35) \cdot 4 = 2.3$

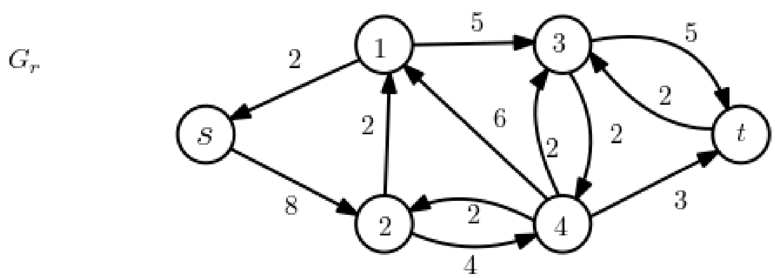
(c) How many different optimal prefix codes are there for part (b)? ***(1 bonus marks)***

Symbols at levels 1, 2, 3 and 4 of code tree are 1, 1, 0, 4, respectively. The last four symbols with common ancestor at level 2, have 4 possible prefixes of length 2. Whatever that is, it determines the codes for the symbol on levels 1 (α) and symbol on level 2 (β). The symbols on level 4 can be in any permutation. Thus, $4 \cdot 4! = 96$ different optimal prefix-codes.

4. Consider the following $s-t$ network with "partial flow / capacities" listed.



(a) Draw the residual digraph. ***(2 marks)***



(b) Find an augmenting path with the largest bottleneck available. ***(1 mark)***

Answer: There is an augmenting path $(s, 2, 4, 1, 3, t)$ with maximum bottleneck 4.

(c) Compute the maximum flow and give a minimum cut as a certificate. (***2 marks)***

Answer: A minimum cut is between nodes $\{s, 2\}$ and $\{1, 3, 4, t\}$ of weight $2 + 6 = 8$.