Name: Rollno:

CS345: Design and Analysis of Algorithms (Quiz 2)

27th October 2024

Total Number of Pages: 4 Time: 1 hr Total Points 30

Question	Points	Score
1	6	
2	4	
3	3	
4	4	
5	7	
6	6	
Total:	30	

Instructions

1. All questions are compulsory.
2. Answer all the questions in the space provided in the question paper booklet.
3. Use the space provided in the paper for rough work.
4. The symbols or notations mean as usual unless stated.
5. You may cite and use algorithms and their complexity as done in the class.
6. Cheating or resorting to any unfair means will be severely penalized.
7. Superfluous and irrelevant writing will result in negative marking.
8. Using pens (blue/black ink) and not pencils. Do not use red pens. for answering.

Helpful hints

1. It is advisable to solve a problem first before writing down the solution.
2. The questions are *not* arranged according to the increasing order of difficulty.

Name: Rollno:

Question 1. (a) (2 points) Given a residual network of a network with real edge weights, how can we find an augmenting path from source to sink in the residual network?

> **Solution:** Run BFS Algorithm from source and see if sink is reachable, and compute the shortest such path.

(b) (2 points) If n is the number of vertices, m is the number of edges, and C is the maximum weight of an edge in the residual network, then what is the time complexity of finding an augmented path from source to sink in the residual network?

(b) $O(m)$

(c) (2 points) Given a network with real edge weights, let f and f' be the flow computed by the max-flow algorithm, before and after the i-th iteration. Let $\delta_f(s,x)$ be the shortest length path from the source s to x in the residual network with respect to the flow f. What is the relation between $\delta_f(s,x)$ and $\delta_{f'}(s,x)$, for any vertex x in the network?

(c) $\delta_{f'}(s,x) \geq \delta_f(s,x) + 2$ **OR** $\delta_{f'}(s,x) \geq \delta_f(s,x)$

Question 2. Consider the Floyd-Warshall algorithm to compute the shortest path between every pair of vertices in a graph.

(a) (3 points) What is the recurrence relation for $D_k(i,j)$ (i.e. the length of the shortest path from i to j with intermediate vertices of index at most k)? Give the base case as well.

> **Solution:** Base Case ($k=0$): $D_0(i,j) = w(i,j)$ if (i,j) is an edge and $D_0(i,j) = \infty$ otherwise.
> For $k > 0$, $D_k(i,j) = \min(D_{k-1}(i,j), D_{k-1}(i,k) + D_{k-1}(k,j))$

(b) (1 point) What is the additional space required to execute Floyd-Warshall algorithm?

(b) $O(n^2)$

Question 3. (3 points) Consider the flow with lower bound problem. You are given a network $G = (V,E)$, a capacity function $c : E \to \mathbb{R}^+$ and a lower bound function $l : E \to \mathbb{R}^+$. You need to compute a flow f such that, for every vertex $v \in V$, $f_{in}(v) - f_{out}(v) = 0$, and for every edge $e \in E$, $l(e) \leq f(e) \leq c(e)$.

Give a reduction from the flow with lower bound problem to the circulation with demand problem. For this you need to define the new network $H = (V_H, E_H)$, a capacity function c^* and a demand function d^* for H.

> **Solution:** H is defined as follows, $V_H = V$ and $E_H = E$.
>
> For all edges e, $c^*(e) = c(e) - l(e)$.
>
> For all v, $d^*(v) = \sum_{(v,z)\in E} l(v,z) - \sum_{(u,v)\in E} l(u,v)$

Question 4. Let ϕ be the potential function associated with an amortized analysis of a sequence of n operations.

(a) (2 points) What is the amortized cost of the i-th operation?

(a) **Actual cost of i-th operation +** $\phi(i) - \phi(i-1)$

(b) (2 points) Show that the actual cost of all n operations is upper bounded by the amortized cost of the n operations.

Solution:

$$\begin{aligned}
\text{amortized cost of all } n \text{ operations} &= \sum_i \text{amortized cost of } i\text{-th operation} \\
&= \sum_i (\text{actual cost of } i\text{-th operation} + \phi(i) - \phi(i-1)) \\
&= \text{actual cost of all } n \text{ operations} + \phi(n) - \phi(0) \\
&= \text{actual cost of all } n \text{ operations} + \phi(n) \\
&\geq \text{actual cost of all } n \text{ operations}
\end{aligned}$$

Question 5. In class we saw an amortized analysis of counting the number of bit flips, under n increment operations. Now suppose we also allow the decrement operation, that is reducing the current binary counter by 1. Consider a sequence S of n operations, such that each operation is either *increment* or *decrement*.

(a) (2 points) What is the worst case upper bound on the number of bit flips for executing S?

(a) $O(n \log n)$

(b) (2 points) Give an analysis to prove the upper bound in Part (a).

Solution: The binary counter having a maximum of n increments has at most $\log n$ bits. With n increment/decrement operations the number of bit flips is therefore at most $O(n \log n)$.

(c) (3 points) Show that the bound in Part (a) is tight. In other words give a sequence S, for which the number of bit flips matches the bound in Part (a)

Solution: Consider a sequence having about $n/2$ "increment" operations to get the binary number $100\ldots00$. Now for the remaining $n/2$ operations alternate between "decrement" and "increment". The number of bit flips for each such operation in $\log n$. Hence the total number of bit flips for this sequence is at least $n/2 \log n$.

Question 6. Consider the Move to Front (MTF) algorithm discussed in class for the Online List Search problem. Let OPT be the optimal algorithm for this problem. Let $r(x)$ and $r^*(x)$ be the ranks of an element x in the MTF list and the OPT list respectively.

(a) (1 point) The competitive ratio of MTF is **4**.

(b) (1 point) The potential function $\phi(i)$ in the competitive ratio analysis is 2·**(no. of inversions)**.

(c) (2 points) When the MTF algorithm moves an element to the front of the list, what is the number of new inversions created?

(c) $\leq r^*(x) - 1$

(d) (2 points) When the MTF algorithm moves an element to the front of the list, what is the number of old inversions that are destroyed?

(d) $\geq r(x) - r^*(x)$