Habib University

CS 412 (Algorithms: Design and Analysis)

Spring 2024 – Quiz 01 – L2

January 25, 2024. Total Marks: 5. Duration: 25 minutes.

---

1. (1 point) Consider two functions $f = 2^n$ and $g = 2^{n/2}$. Is $f(n) = \Omega(g(n))$?
2. (2 points) Show that $\Theta(f(n)) + \Theta(g(n)) = \Theta(f(n) + g(n))$
3. (1 point) Is $1/(n^3 + 1) = \omega(1/(1 + n))$?
4. (1 point) Suppose you have two different algorithms A and B whose time complexity is given by the functions $\mathcal{O}n\log(n)$ and $\mathcal{O}n^{1.5}$ respectively. Which algorithm is slower asymptotically?

Habib University

CS 412: Design and Analysis of Algorithms

Spring 2024 – Quiz 02

February 15, 2024. Time: 25 minutes. Total points: 05.

1. **[1 point]** Argue that Merge sort is an asymptotically optimal comparison-based sorting algorithm.

2. Consider a sequence of n elements $\langle a_1, a_2, \dots, a_n \rangle$, $n \geq 1$.

   a. **[2 points]** Write the pseudocode of a divide-and-conquer algorithm that takes into input a sequence of n elements and returns its maximum. Note that when $n = 1$, a_1 is the maximum. Assume n is in the exact power of 2.

   b. **[0.5 points]** Write a recurrence relation for the above divide-and-conquer algorithm with the base case.

   c. **[0.5 points]** Solve the above recurrence relation using the Master theorem.

> **Master theorem**[2] *If* $T(n) = aT(\lceil n/b \rceil) + O(n^d)$ *for some constants* $a > 0$, $b > 1$, *and* $d \geq 0$, *then*
>
> $$T(n) = \begin{cases} O(n^d) & \text{if } d > \log_b a \\ O(n^d \log n) & \text{if } d = \log_b a \\ O(n^{\log_b a}) & \text{if } d < \log_b a\,. \end{cases}$$

Source: Vazirani et al.

3. **[1 point]** Solve the following recurrence: $a_n = a_{n-1} + 2a_{n-2}$, with $a_0 = 2$ and $a_1 = 7$ using the method discussed in the class

Habib University

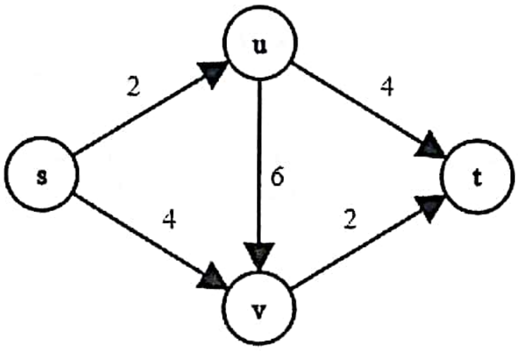
CS 412: Design and Analysis of Algorithms

Spring 2024 – Quiz 03 – L2

March 14, 2024. Time: 25 minutes. Total points: 05.

1. **[2 points]** Let f be a flow in a flow network $G = (V, E)$ with a source s and a sink t. Argue that $|f| = c(S,T)$, for some $cut(S,T)$ of $G \Rightarrow f$ is a maximum flow in G.

2. **[2 points]** Consider the Consider the following directed graph G. Find the min-cut using the Ford–Fulkerson method.



3. **[1 point] True/False.** When the Ford-Fulkerson method terminates, each back-flow edge from v back to u in the residual graph represents the final flow value for edge (u, v) in the flow graph. Justify your choice in one to two sentences only.

Habib University

CS 412: Design and Analysis of Algorithms

Spring 2024 – Quiz 04 – L2

April 4, 2024. Time: 25 minutes. Total points: 05.

1. **[3 points]** An old public library wants to digitize its book cataloging process, with the goal of optimizing the storage and retrieval system. The books need to be stored in a way that maximizes the use of limited shelf space while ensuring that popular books are easily accessible to interested readers. Formulate this as a bottom-up dynamic programming problem. Does the problem exhibit optimal substructure and overlapping subproblems property? Give a pseudocode to solve this problem. What is its runtime complexity?

2. **[1 point]** Given a dag G and a source vertex s, give a linear time solution [in pseudocode] to find the longest distances from s to all other vertices in G.

3. **[1 point]** Prof. Geller claims that when it comes to asymptotic runtime complexity, bottom-up dynamic programming and memorization are not different. Do you dis/agree with this statement? Give your rationale in one to two sentences only.

Habib University

CS 412: Design and Analysis of Algorithms

Spring 2024 – Quiz 05 – L2

April 25, 2024. Time: 25 minutes. Total points: 05.

1. **[2 points]** Find an optimal Huffman code for the string ABRACADABRA. Show complete working.

2. **[1 point]** In which scenario(s), the characteristics of a problem will lead you to prefer dynamic programming over a greedy-based solution?

3. **[2 points]** Imagine a classroom where each of $n \in \mathbb{Z}^+$ students submit a homework assignment, and the absent-minded professor returns the assignments randomly to the students without checking their names. How many students, on average, are likely to receive their own homework back by chance?