Finals Question Bank: Analysis of Algorithms
May 28, 2025

---

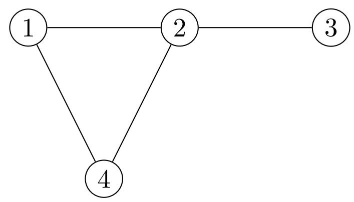
1 Let us consider the random experiment of rolling two dice.

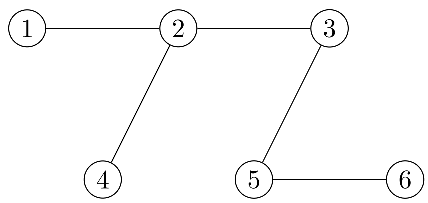
1. Define a random variable X as the number of 6's that you get. Find the expected value of X.
2. Define a random variable Y as the sum of the two dice. Find the expected value of Y.

2 Perform the analysis of Median of Median's Algorithm where we make chunks of size 9 instead of 5 and solve the recurrence using substitution method.

3 Let X be a random variable that is equal to the number of heads in two flips of a fair coin. What is $E(X^2)$? What is $(E(X))^2$?

4 Apply the algorithm step by step to find the articulation points in the following graphs:





5 Given an undirected graph with distinct non-negative edge weights. Suppose that you have computed shortest paths to all nodes from a particular node s. Now suppose each edge weight is increased by 1; the new weights are $w_e := w_e + 1$. Do the shortest paths change? Give an example where they change or prove they cannot change.

6 Give a general example of a weighted, directed graph G on n nodes where at least half of the edge weights are negative, and Dijkstra's algorithm correctly finds the shortest paths from a source vertex. Note that n is not fixed, so you cannot just give an example on 10 or 20 vertices. Your job is to maximize the number of edges that contain negative weights.

7 How can we use the output of the Floyd-Warshall algorithm to detect the presence of a negative-weight cycle?

8 "In GoldExtractor game, you collect coins using two extractors: a red vertical one that moves right and a blue horizontal one that moves up. Each time an extractor moves to a new line containing coins, it collects one coin, and all other coins on that line are lost. The game ends when either extractor reaches its final position, so plan your moves carefully to maximize your total score!"
Design a complete Dynamic Programming algorithm for the problem. Your input is (x,y) coordinates of n coins, and your output is a number which is maximum number of coins collected. You can assume that $0 < x, y < n$.

9 Find the decision version of the minimum spanning tree problem. State the original problem and the decision problem formally.

10 Rod Cutting Problem
Given a rod of length n and an array of prices `price` such that `price[i]` represents the price of a rod of length $i + 1$, determine the maximum value obtainable by cutting up the rod and selling the pieces.

- **Input:** $n = 8$, `price` = [1, 5, 8, 9, 10, 17, 17, 20]
- **Output:** 22 (Cut into rods of lengths 2 and 6 for prices 5 and 17)

11 Prove that the following problem is in NP:
Given an integer x, is x NOT a prime?

12 Reduce 4-SAT to 3-SAT.

13 Reduce Subset Sum to Partition problem.

14 Reduce Partition to Subset Sum problem.

15 Reduce Partition to Bin Packing probem.

16 Reduce Vertex Cover to Set Cover problem.

17 Reduce Maximum Clique to Subgraph Isomorphism problem.

18 Reduce Subset Sum to Knapsack problem.

19 Reduce Vertex Cover to Maximum Dominating Set problem.

20 Solve the following recurrence relations:

a. $T(n) = 2T(n/3) + 1$
b. $T(n) = 5T(n/4) + n$
c. $T(n) = 9T(n/3) + n^2$
d. $T(n) = 2T(n-1) + 1$
e. $T(n) = T(\sqrt{n}) + 1$

21 An array $A[1 \ldots n]$ is said to have a **majority element** if more than half of its entries are the same. Given an array, the task is to design an efficient algorithm to determine whether the array has a majority element, and, if so, to find that element. The elements of the array are not necessarily from some ordered domain like the integers, so comparisons of the form "is $A[i] > A[j]$?" are not allowed. (Think of the array elements as GIF files, for instance.) However, you can answer questions of the form: "is $A[i] = A[j]$?" in constant time.

Part 1: Show how to solve this problem in $O(n \log n)$ time. **Hint:** Split the array A into two arrays A_1 and A_2 of half the size. Does knowing the majority elements of A_1 and A_2 help you figure out the majority element of A? If so, you can use a *divide-and-conquer* approach.

Part 2: Can you give a linear-time algorithm? **Hint:** Here's another *divide-and-conquer* approach:

- Pair up the elements of A arbitrarily to get $n/2$ pairs.
- Look at each pair: if the two elements are different, discard both of them; if they are the same, keep just one of them.

Show that after this procedure there are at most $n/2$ elements left, and that they have a majority element if and only if A does.

22 Write an algorithm to merge three sorted arrays into a single sorted array.

23 Write an algorithm to merge k sorted arrays into a single sorted array.

24 Given two arrays A and B, write an algorithm to find the median of $A \cup B$ in the following cases:

- When A and B are sorted.
- When A and B are unsorted.

25 Give an $O(n \log k)$-time algorithm to merge k sorted lists into one sorted list, where n is the total number of elements in all the input lists. (Hint: Use a min-heap for k-way merging.)

26 You are given an array of strings, where different strings may have different numbers of characters, but the total number of characters over all the strings is n. Show how to sort the strings in $O(n)$ time. (Note that the desired order here is the standard alphabetical order; for example, $a < ab < b$.)

27 What is the running time of the Breadth-First Search (BFS) algorithm if we represent its input graph using an adjacency matrix and modify the algorithm to handle this form of input?

28 The **diameter** of a tree $T = (V, F)$ is defined as $\max\{\delta(u, v) : u, v \in V\}$, that is, the largest of all shortest-path distances in the tree. Give an efficient algorithm to compute the diameter of a tree, and analyze the running time of your algorithm.

29 For each of the following strings:

a. Build a frequency table for the characters.

b. Construct the Huffman merge tree.

c. Generate Huffman codes for each character.

d. Encode the full string using the Huffman codes.

e. Compare the number of bits in the Huffman-encoded version with standard ASCII encoding (8 bits per character).

`AGCTTAGGCTTAA`

30 Coin Change Problem

Given coin denominations $c_1, c_2, \ldots, c_n$ and a target amount A, find the minimum number of coins needed to make change for the amount A. Assume an infinite supply of coins of each denomination.

31 Longest Increasing Subsequence

Given a sequence of integers $A = a_1, a_2, \ldots, a_n$, find the length of the longest subsequence such that all elements of the subsequence are sorted in increasing order.

32 Reduce Maximum Independent Set to Maximum Clique problem.

33 Reduce Hamiltonian Cycle to Hamiltonian Path problem.