Sample Problems and Solutions
Algorithms
State University of New York at Stony Brook
Instructor: Prof. Pramod Ganapathi

[Decrease-and-conquer.]

1. Show step-by-step details of how to partition the array $[3, 7, 5, 4, 2, 8, 1, 5, 6, 4, 2, 8]$ using the first element of the array as the pivot element using:
   (a) [5 points] Lomuto partition
   (b) [5 points] Hoare partition

2. [5 points] For the directed acyclic graph as shown in Figure 1, list all possible topological sorts.

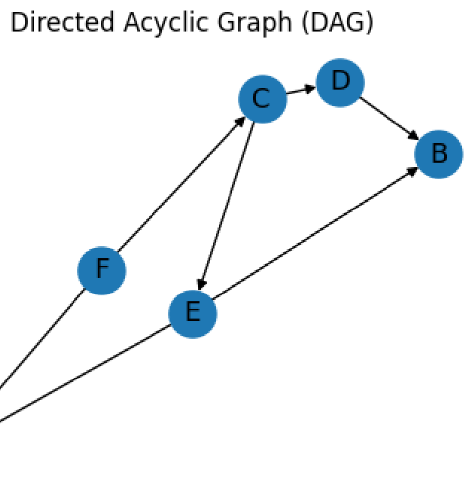


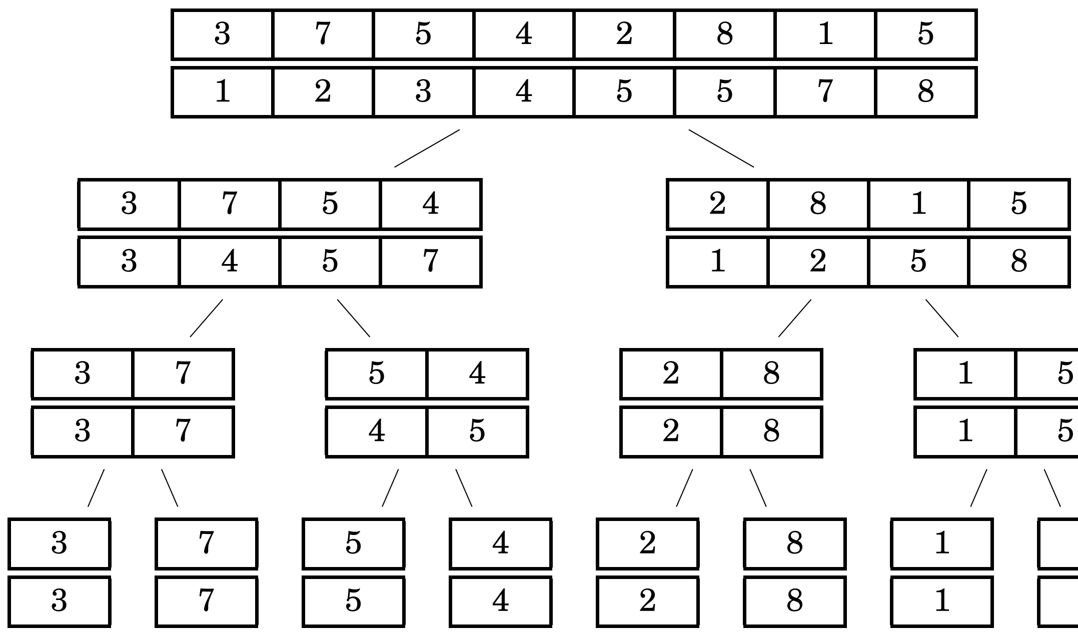
Figure 1: A directed acyclic graph.

[Divide-and-conquer.]

1. Write the divide-and-conquer diagram to sort array $[3, 7, 5, 4, 2, 8, 1, 5]$ using:
   (a) [5 points] Merge sort algorithm
   (b) [10 points] Quicksort algorithm

   Solution:

   (a) Merge sort:



[Greedy technique.]

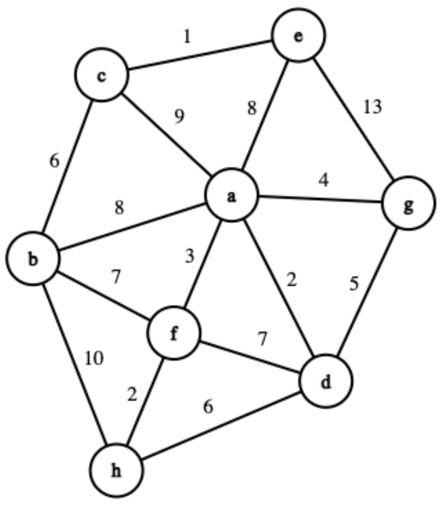
1. Consider Figure 2. The left part of the diagram shows an undirected weighted graph. The right part of the diagram shows the partial execution of Jarnik/Dijsktra's algorithms using a table. Your aim is to fill the table (which shows the complete execution) using the two greedy algorithms: Jarnik and Dijkstra. Each column corresponds to a vertex. Each row corresponds to a vertex that is added to the expanding tree (MST tree or shortest path tree) in each iteration. The single highlighted cell in each row represents the greedy choice for the next iteration. If there are multiple choices that have the minimum distance, choose the vertex that is alphabetically first. For both algorithms we start from vertex a.

   (a) [10 points] Fill the table using the Jarnik's algorithm, where each cell in the table has an ordered pair $(distance, parent)$, where $distance$ represents the shortest distance of the corresponding vertex from the evolving tree and $parent$ represents the parent vertex.

   (b) [10 points] Fill the table using the Dijkstra's algorithm, where each cell in the table has an ordered pair $(distance, parent)$, where $distance$ represents the shortest distance of the corresponding vertex from the source vertex and $parent$ represents the parent vertex.

[Dynamic programming.]

1. [10 points] Given two strings $x[1 \dots m]$ and $y[1 \dots n]$, the goal is to compute the length of the longest common subsequence (LCS) between x and y. Write the subproblem, recurrence, algorithm code, and construct the dynamic programming table for strings $x =$ einstein and $y =$ newton.



Iter.	Vertex added	a	b	c	d	e	f	g	h
0	$-$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$	$(\infty, -)$
1	a	$(0, -)$	$(8, a)$	$(9, a)$	$(2, a)$	$(8, a)$	$(3, a)$	$(4, a)$	$(\infty, -)$
2									
3									
4									
5									
6									
7									
8									

Figure 2: Left: An undirected weighted graph. Right: Table for showing the execution of greedy algorithms.

2. [10 points] Given two strings $x[1 \dots m]$ and $y[1 \dots n]$, the goal is to compute the minimum number of edits required to convert string x to string y using the three operations: insert, remove, and replace. Write the subproblem, recurrence, algorithm code, and construct the dynamic programming table for strings $x =$ einstein and $y =$ newton.

3. Given a set of n items numbered from 1 to n such that item i has weight $w[i]$ and value $v[i]$, the goal is to find the maximum total value of the items that can fit in a knapsack with the maximum weight capacity of W in two cases:

   (a) [10 points] [0-1.] each item can be considered only once.

   (b) [10 points] [Unbounded.] each item can be considered ∞ times.

   For each case, write the subproblem, recurrence, algorithm code, and construct the dynamic programming table for $n = 5$, weights $w[1 \dots 5] = [2, 3, 7, 9, 15]$, values $v[1 \dots 5] = [10, 15, 25, 13, 55]$, and capacity $W = 12$.