Problem 1:  \( (7 + 7 + 6 = 20 \text{ Points}) \)
This question is about making the binary search dynamic. Binary search of a sorted array takes logarithmic search time, but the time to insert a new element is linear in the size of the array. We can improve the time for insertion by keeping several sorted arrays.

Specifically, suppose that we wish to support \textsc{Search} and \textsc{Insert} on a set of \( n \) elements. Let \( k = \lceil \lg(n + 1) \rceil \), and let the binary representation of \( n \) be \( \langle n_{k-1}, n_{k-2}, \ldots, n_0 \rangle \). We have \( k \) sorted arrays \( A_0, A_1, \ldots, A_{k-1} \), where for \( i = 0, 1, \ldots, k - 1 \), the length of array \( A_i \) is \( 2^i \). Each array is either full or empty, depending on whether \( n_i = 1 \) or \( n_i = 0 \), respectively. The total number of elements held in all \( k \) arrays is therefore \( \sum_{i=0}^{k-1} n_i 2^i = n \). Although each individual array is sorted, there is no particular relationship between the elements in different arrays.

a. Describe how to perform the \textsc{Search} operation for this data structure. Analyze its worst-case running time.

b. Describe how to insert a new element into this data structure. Analyze its worst-case and amortized running times.

c. Discuss how to implement \textsc{Delete}.

Problem 2:  \( (4 + 4 + 4 + 3 = 15 \text{ Points}) \)
Solve Exercise 3.3 from DPV. Note that \textit{pre} and \textit{post} numbers are respectively \textit{start} and \textit{finish} times.

Problem 3:  \( (10 \text{ Points}) \)
Solve Exercise 3.11 from DPV.

Problem 4: \textbf{Undirected vs. directed connectivity}  \( (5 + 5 + 5 = 15 \text{ Points}) \)
Solve Exercise 3.13 from DPV.

Problem 5: \textbf{Prim and Kruskal}  \( (10 + 10 = 20 \text{ Points}) \)
Solve Exercise 5.2 from DPV.

Problem 6: \textbf{Dijkstra}  \( (12 + 8 = 20 \text{ Points}) \)
Solve Exercise 4.1 from DPV.