Homework Assignment #2

Problems

1. Answer the following questions about optimal binary search trees (OBST).
   (a) Devise a way to compute the sums $\sum_{i=low}^{high} p_i$ $(1 \le low \le high \le n)$ in constant time per sum. If you need to have a pre-processing step before the computation of sums, the cost of the pre-processing step must be $\mathcal{O}(n)$ in time and space.
   (b) "The root of an OBST always contains the key with the highest frequency." Is this statement true or false? Justify your answer.
   (c) How would you construct an OBST for a set of n keys if all the keys are equally likely to be searched for? What will be the average number of comparisons made by a successful search in such an OBST?
2. Exercises 9.14 and 9.15 in page 337. [Closed hashing with linear/quadratic probing]
3. Build a linear hash table by inserting the following eleven keys 12, 44, 13, 88, 23, 94, 11, 39, 20, 16, 5 in the given order. The hash table is initially empty and has eight slots. Use a family of hash functions $h_k(x) = x \bmod 2^k$.
4. Show the result of inserting 3, 1, 4, 6, 9, 2, 5, 7 into an initially empty max-heap. Draw a max-heap (*i.e.*, a binary tree) resulted from each insertion.
5. You are given eight integers stored in an array: 3, 1, 4, 6, 9, 2, 5, 7. Heapify this array to build a max-heap using an $\mathcal{O}(n)$ algorithm. Show the array resulted from each swapping of array elements. Drawing actual trees is not necessary.
6. Give an algorithm that, given an undirected graph $G = (V, E)$ stored in an adjacency list, outputs "CYCLIC" if G contains a cycle, and outputs "ACYCLIC" otherwise. Your algorithm should run in $\mathcal{O}(\,|V|\,)$ time. Explain why your algorithm takes $\mathcal{O}(\,|V|\,)$ time.
7. Exercise 11.1 in page 399. [Induction on the number of edges]
8. Exercises 11.4 and 11.6 in page 400. [DFS and BFS]
9. Find a topological order for a graph in Figure 11.16 (page 390). You can ignore the weights on the edges of the graph.
10. Exercise 11.10 in page 400. [Dijkstra's algorithm]
11. Run the Bellman-Ford algorithm on the graph of Figure 11.26 beginning at vertex 4
    (a) after changing the weight of (3,5) from 15 to -15.
    (b) after changing the weights of (2,4) and (3,5) from 5 and 15 to -5 and -15, respectively.
12. Exercise 16.3 in page 532. [Floyd's algorithm]
13. Exercise 11.17 in page 401. [Prim's algorithm]
14. Exercise 11.18 in page 401. [Kruskal's algorithm]
15. Give an algorithm that finds a *maximum* spanning tree. (Exercise 11.19 in page 401)
16. When can Prim's and Kruskal's algorithms yield different MSTs? (Exercise 11.20 in page 401)
17. Prove that if all edge weights are distinct in a graph G then there exists only one MST in G. (Exercise 11.21 in page 401)

Due date

There is no due date for this homework assignment because it will not be graded. The purpose of this homework assignment is to provide students with additional opportunities to review the materials covered in the class.