CS 210 Advanced Data Structures & Algorithms
Fall 2020

Homework Assignment #2

(Due: 10/29/20)

[100 points]

Part 1: Written Exercises

1. **[10 points]** ***(Greedy Approach)*** **Chapter 4: Exercise #2 & #7** and **#12**
2. **[10 points]** ***(Backtracking)*** **Chapter 5: Exercise #2 (**The **5-Queens** Problem) **and #18** (No need to show all the actions step by step. Draw the *pruned* state space tree *via* ***Backtracking*** up to the point where the *first* solution is found and show the first solution.)
3. **[5 points]** ***(Branch-and-Bound)*** **Chapter 6:** Draw the *pruned* state space tree via ***Best-First Search B&B*** and show the optimal solution of the 0-1 Knapsack Problem: *W = 160 lb & Item1= $40 & 20 lb, Item2= $30 & 50 lb, Item3= $50 & 100 lb, Item4= $10 & 50 lb*.
4. **[10 points]** ***(Computational Complexity: The Sorting Problem)*** **Chapter 7:**
    a. What is the worst-case time complexity of **HeapSort** using **ternay heap**? Explain.
    b. What is the **lower bound for sorting** *n* items only by comparison of keys? Explain.
5. **[15 points]** ***(Advanced Heaps)***
    a. What is the worst-case time complexity of the **top-down binary heap building**? What is the worst-case time complexity of the **bottom-up binary heap building**? Explain.
    b. Draw the **leftist min-heap** that results when you insert items with the keys 77, 22, 9, 68, 16, 34, 13 and 8 in that order into an initially empty leftist min-heap.
    c. Draw the **skew min-heap** that results when you insert items with the keys 77, 22, 9, 68, 16, 34, 13 and 8 in that order into an initially empty skew min-heap.

Part 2: Programming Exercises*

1. **[20 points]** ***(Greedy Approach)*** **Greedy Approach-based Dijkstra Algorithm for the Single Source Shortest Path Problem (Chapter 4: Exercise #14)**

   You should output the distance array **D**, the path array **P** and the **shortest paths**. Test your implementation using (1) the test case given in class and (2) your own test cases including digraphs with *non-negative* edges only and digraphs with some *negative* edges.

2. **[30 points]** ***(Advanced Heaps)*** **The Leftist Min-Heap**

   Design and implement the Leftist Min-Heap ADT. You should include ***merge, insert, deleteRoot, showLH*** and ***showSPL***.

   The operation ***showLH*** prints the *priority values* in the leftist min-heap as rotated counterclockwise 90 degrees from its conventional orientation using a "reverse" inorder tree traversal.

   The operation ***showSPL*** prints the *spl values* in the leftist min-heap as rotated counterclockwise 90 degrees from its conventional orientation using a "reverse" inorder tree traversal.

   Test your implementation using (1) the test case given in class and (2) your own test cases via *showLH* and *showSPL*.

*** NOTE:**

- Programming in C++ using Visual Studio
- **Pair Programming**
- Deliverables:
  - ✓ **The screenshots of test results for all test cases.**
  - ✓ **The Visual Studio project folder.**

*** End of Homework Assignment #2 ***