

Assignment 5

(Due Friday of week 16 - February 15)

Complete the below exercises:

- a) **CompareKnapsack:** Discuss the differences between the solution for knapsack problem using Backtracking (Algorithm 5.7) versus Branch and Bound (Algorithm 6.2).
- b) **Mergesort:** Write solution to the below exercise in the programming language you selected in A1. Program file should have your name at the top in comment, with short description of what is implemented in that file. Make sure your file(s) have appropriate names as indicated in each exercise. Program should write output to the Console and have hard coded input in main.

Note 1: If a program is not in approved programming language (or in different language than previous assignment) or has any syntax error, no points will be awarded for that exercise

Note 2: Submitting wrong files or in the wrong format or corrupted files will not be accepted nor will any re-submission be allowed for any such mistake.

Implement Linked Version mergesort from chapter 7 (Algorithm 7.4). Compare the time complexity to the algorithm you did for Assignment 3 (based on given pseudo code). **Use a few instances of different data sets to run against each implementation and compare the runtime results.** Discuss time complexity of both algorithms and how they compare to each other relative to your runtime results.

- c) **CompareSearch:** List and discuss the advantages and disadvantages of all the searching algorithms discussed in chapter 8 (binary search tree, interpolation search, b-tree, selection, and hashing) and the improvements suggested in the chapter. For each of the searching algorithms give two examples of situations in which that algorithm is the most appropriate.
 - d) **CompareDesign:** In your own words discuss the differences and similarities between Backtracking versus Branch and Bound approach design approaches. What type of problems are well suited or not for each approach and why. Give specific examples to support your discussion. Also discuss how these compare to the other three approaches you learned (D&C, dynamic, greedy). The comparison should be about two pages in length (max 12-point font).
2. Record a video about 15-20min long explaining the comparisons, explaining your program implementation, test program and its output.
 3. **Submission instructions:** Submit one file (**Word or PDF**) with comparison and time/space complexity discussions; second file (rar or zip) with the code; and third file with the zip/rar of video

Grading Rubric

NOTE: You must submit the video to receive any credit for the comparisons and programs. Both the correctness of the programs/compare and the correctness of your explanation are graded from the video.

Points	Criteria
5	Programs use object oriented program approach. Programs have the appropriate naming convention, author's name, and brief description of the implementation in each file
20	CompareKnapsack: Discussion includes pros and cons knapsack solution using Backtracking (Algorithm 5.7) versus Branch and Bound (Algorithm 6.2). Discussion includes time/space complexity comparison of each Video summarizes the discussion and conclusions
25	Mergesort: Correct algorithm of mergesort (Linked Version Algorithm 7.4) is implemented correctly There is test code with at least one data set to run against the algorithm and capture runtime There is comparison of the time complexity of 7.4 algorithm to assignment 3 algorithm Analysis documents what data were used and the runtime results for each algorithm There is discussion of time complexity of both algorithms and how they compare to each other relative to your runtime results Video explains implementation of 7.4 algorithm, runs the test program, and explains output Video summarizes the comparison and analysis and conclusions
25	CompareSearch: There is a discussion of the advantages and disadvantages of all the searching algorithms discussed in chapter 8 (binary search tree, interpolation search, b-tree, selection, and hashing) and the improvements suggested in the chapter. For each of the searching algorithms there are two examples given of the situations in which that algorithm is the most appropriate. Video gives highlights of advantages and disadvantages and discusses the two examples for each algorithm
25	CompareDesign: The compare and contrast is written in student's own words, represents graduate level depth and breath, has at least 50% original thought, uses appropriate examples to support discussion, and does not plagiarize any of the materials.