Assignment 4
(Due Friday of week 13 - Jan 25)

Complete the below exercises:

a) **Compare:** In your own words discuss the differences and similarities between Divide and Conquer, Dynamic Programming, and Greedy Approach design approaches. What type of problems are well suited or not for each approach and why. Give specific examples to support your discussion. The comparison should be about one to two pages in length (max 12-point font).

b) **Analysis1 (D&C):** Analyze the complexity of algorithm A which solves a problem of size n by dividing it into 4 subprograms of size n/4, recursively solves each subprogram, and combines in $O(n^2)$ time. Determine the recurrence and whether it is "Subtract and Conquer" or "Divide and Conquer", and then solve it to big O notation. Use the master theorem to solve and show the work. In the video you must explain how you obtained the recurrence from the problem description and show how you solved step by step using the master theorem we went over in class.

c) **Analysis2 (D&C):** Analyze the complexity of algorithm B which solves a problem of size n by recursively solving one sub-program of size n-2 and combines in $O(n^2)$ time. Determine the recurrence and whether it is "Subtract and Conquer" or "Divide and Conquer", and then solve it to big O notation. Use master theorem to solve and show the work. In the video you must explain how you obtained the recurrence from the problem description and how you solved step by using the master theorem we went over in class.

d) **Floyd (Dynamic prog):** In your own words explain the Floyd algorithm in your textbook's (p. 108 A3.4) that we went over in class. Then using Fig 3.2 (p. 101) graph below and the array P solution in Fig 3.5 (p.108) also below, discuss the shortest distance and path between nodes v2 and v1 and the intermediate vertices.

Figure 3.2
A weighted, directed graph.

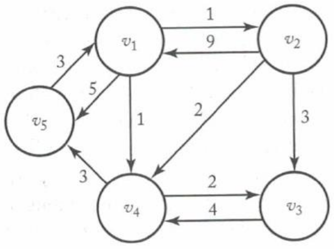
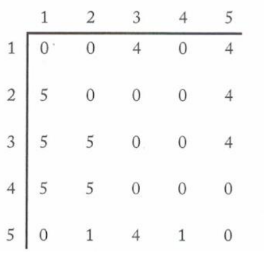


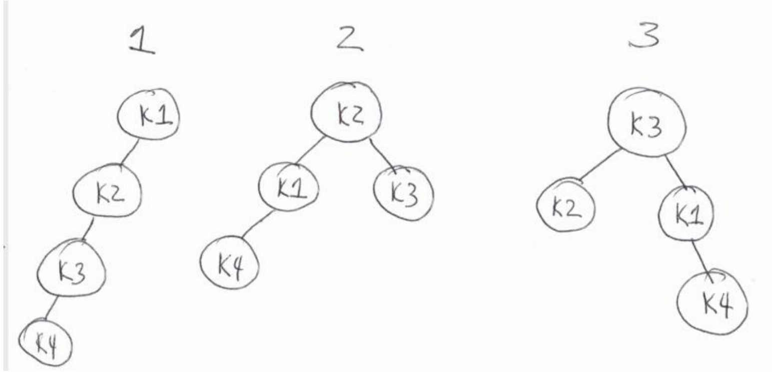
Figure 3.5
The array P produced when Algorithm 3.4 is applied to the graph in Figure 3.2.



	1	2	3	4	5
1	0	0	4	0	4
2	5	0	0	0	4
3	5	5	0	0	4
4	5	5	0	0	0
5	0	1	4	1	0

e) OptimumBinarySearchTree: Given the below trees and probabilities determine which is the optimum tree. Show the calculations.

P1=0.2, P2=0.1, P3=0.4, P4=0.3



f) 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.

Schedule (Greedy) Implement algorithm 4.4 (pp., 179-180) and solve the below problem instance. What is the final value of feasible sequence and profit?

Job	Deadline	Profit
1	2	46
2	4	52
3	3	30
4	3	36
5	2	56
6	1	40

2. Record a video about 15min long explaining the analysis solutions, floyd algorithm answers, optimum binary search tree solution, and schedule algorithm implementation, analysis, and output.

3. **Submission instructions:** Submit one file (**Word or PDF**)with comparison of algorithm design approaches; second file with solutions to Analysis 1 and 2 and optimum binary search tree; third file with program for scheduling problem; and fourth file with the zip/rar of video

Grading Rubric

NOTE: You must submit the video to receive any credit for the programs. Both the correctness of the programs/analysis and the corrected 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
25	**Compare:** 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.
10	**Analysis1:** Algorithm A: Gives correct recurrence explaining how it was obtained and whether it is "Subtract and Conquer" or "Divide and Conquer"; Solves showing step by step work and presents solution in Big O notation Video correctly explains the analysis of algorithm A and how recurrence was determined; how it was solved using the master theorem (which theorem) step by step, and what the time complexity is in Big O notation
10	**Analysis2:** Algorithm B: Gives correct recurrence explaining how it was obtained and whether it is "Subtract and Conquer" or "Divide and Conquer"; Solves showing step by step work and presents solution in Big O notation Video correctly explains the analysis of algorithm A and how recurrence was determined; how it was solved using the master theorem (which theorem) step by step, and what the time complexity is in Big O notation
15	**Floyd:** Video correctly and thoroughly explains Floyd shortest path algorithm 3.4 Video correctly and thoroughly explains the array P solution (Fig 3.5) for graph (Fig 3.2) and discusses the shortest distance and path between nodes v2 and v5 and the intermediate vertices.
15	**OptimumBinarySearchTree:** Correctly shows the calculations to determine optimum tree Gives and explains in the video the correct solution
20	**Schedule:** Correctly implements the algorithm from the textbook A4.4 (as went over in the class) Correctly gives the final value and profit Video explains the implementation, shows the program running, and explains the output