

Data Structure: Assignment 4

Due: 2019/1/14 (Monday)

- Requirement: Implementation with C/C++ program
 - Total: 100 points (**50 points** for each question)
-

◆ Submission instructions:

- [1]. Write a “**README file**” including the answers to problems 1-2, and a detailed note about the functionality of each of the above programs, and complete instructions on how to run them.
 - [2]. Make sure you include your name in each program and in the README file.
Each question has one C/C++ program file and one README file. Make sure all your programs are fully commented, and compile and run correctly on the Linux-based machines.
 - [3]. Submit your assignment to the portal system by the due date.
-

1. Given a directed, weighted graph G with n vertices (V_1, V_2, \dots, V_n) , let the set of paths from vertex V_1 to vertex V_n be $P = \{P_1, P_2, \dots, P_m\}$, find out the 3 shortest paths, P_i, P_j, P_k , in P .

- (1). Write a program to find out the 3 shortest paths.

Input format description:

- The first line contains a number n , indicating the number of vertices
- Each line from the second to the last line contains three numbers:
 $V_i V_j W_{ij}$, indicating an edge from vertex V_i to vertex V_j , with a weight W_{ij}

Output format description:

- Displays each of the 3 paths on screens, including the vertices along the path and the total weight of the path.

- (2). Analyze the time complexity of your program in the readme file.

2. We only want to obtain the i^{th} smallest of n unsorted elements rather than sort all input elements. Please adapt quicksort to achieve this purpose. Note that you do not need to sort all input elements by quicksort.

- (1). Write a program to resolve this problem.

Input format description:

- The first line contains an unsorted sequence S
- The second line contains the value of i .

Output format:

- Print the i^{th} smallest element in the given unsorted sequence S

- (2). Analyze the time complexity of your program in the readme file.