

CS 211 Data Structures and Algorithms Lab
July -- December, 2018
Assignment 1
Total Marks: 10
Due on 4th August

A graph is a discrete structure with two types of elements - vertices (nodes) and edges. An edge connects two vertices. You can think it as a road network where junctions correspond to vertices and roads correspond to edges.

An edge e is incident to a vertex v if e connects v with another vertex. In a graph G , the degree of a vertex v is the number of edges incident to v .

Input format: The input is a file containing the description of a graph -- we call them graph files. Every line in the graph file contains two numbers separated by a whitespace. A line in the file represents an edge in the graph where the two numbers in the line are the labels of the vertices to which the edge is incident. For example, a file with the following three lines represents a path of four vertices:

```
1 2
2 3
3 4
```

It is not necessary that the labels of the vertices are numbers. The label of a vertex can be any combination of letters, digits and `_`.

Task 1 (marks: 2): Read an input graph file. Create an output file in the following format: The first line should contain the number of vertices and the number of edges separated by a whitespace. The rest of the lines should contain all the labels of the vertices - one in each line, in any order. For example, an output of task 1 for the above mentioned input file will be:

```
4 3
1
2
3
4
```

Task 2 (marks: 3): Create an output file, where every line corresponds to a vertex and its degree (in any order). For example, for our input, an output of task 2 will be:

```
1 1
2 2
3 2
```

4 1

Task 3 (marks: 5): Create an output file where the output of task 2 is sorted based on the degree (in ascending order) using insertion sort. In the above case, an output will be:

1 1

4 1

2 2

3 2

Submission and Evaluation

The program you submit should output task1.out, task2.out and task3.out when run. Input file will be given to the program as a command-line argument. The main file of your program should be named as main.<extension>, where the extension depends on the language you choose (Usage of C/C++ is recommended. In case you want to use some other language, please let the instructor know it in advance). Test well before submission. We have some hidden inputs with us to test your program. The mark you obtain is purely based on whether your program correctly gives outputs for the hidden inputs. So, it is not required that you submit the output files. It is important that you follow the input/output conventions exactly (including the naming scheme) as we may be doing an automated evaluation.

This assignment is due on 4th August. Penalty for late submission is 5% per week; i.e., if you submit on 13th August, you will get only 90% of the mark you deserve otherwise.

Follow some coding style uniformly. Provide proper comments in your code.

Submit only through moodle. Submit well in advance. Any hiccups in the moodle/internet at the last minute is never acceptable as an excuse for late submission.

Acknowledge the people (other than the instructor and TAs) who helped you to solve this assignment. The details of the help you received and the names of the people who helped you (including internet sources, if applicable) should come in the main file or in a separate file (acknowledge.txt). Copying others' programs is a serious offence.

Honesty policy of the institute will be strictly followed.

Sample input and output files are attached. This is only for testing purpose.