Institute for Advancing Intelligence (IAI), TCG-CREST
End-Semesteral Examination
Ph.D Program Session: 2022–2023
Subject: Introduction to Computer Programming and Data Structures

Date: 20. 12. 2022 Full Marks: 60 Time : 4 Hours

Instructions:

- Answer any 3 questions.
- Read the questions carefully. Plan your approach. Do necessary rough-work. The pages for rough-work must be submitted. There will be a bonus of 5 marks for the same. However, one can get maximum 60.
- Some of the questions require files. You can download them from the digital version of the question paper kept in the *course webpage*.
- For submission, each solution file must have a name *ES0x_firstName.c*. Keep all such solution in a zip file with name *endsem_submission_firstName.zip*, and send them to *laltu.sardar@outlook.com* with subject as "*endsem_submission_firstName*".

1. Problem id #ES01: Count triangles.
   Let A be the adjacency matrix representation of the graph. If we calculate A^3, then the number of triangles in an undirected graph is equal to $trace(A^3)/6$. Where $trace(A)$ is the sum of the elements on the main diagonal of the matrix A. Given the undirected graph `input_ES01.txt`, with 4039 vertices, count the number of triangles. The number of 2-D matrices you are allowed to take is 2.

   - Input: Sample input files: `input_ES01.txt`
   - Execution: You should run the code as `./a.out`
   - Output format: `triangle_count`.
   - Hint: prepare the adjacency matrix, find multiplication and finally compute trace.
   - Warning: Use `free` function wisely, if needed.

   [20]

2. Problem id #ES02: maximum frequent length
   A list of strings given in `input_ES02.txt`. A string is a *word* if it contains only {A,B,...,Z, a,b,...,z,_}. *Write a program to find the word-length (the number of characters in a word) that occurs most frequently in the input, along with its frequency.*

   For example, if your input consists of

- 17 words that are 3 characters long
- 24 words that are 4 characters long
- 2 words that are 5 characters long
- 10 words that are 2 characters long

Then output will be `4 24`.

Explanation: If the strings in the input files are `xyz a http:t cd foo b#c bar jkl x_yz uvw p_q mnzc b i a*v`. Then, the input has 3 words of length 1, 1 word of length 2, 6 words of length 3, and 2 words of length 4, other strings are not words. So, the output will be `3 6`.

- Input: A file `input_ES02.txt`. Each line of the file is a string having maximum length 30.
- Output: `maximum_frequent_word_length its_frequency`

[20]

3. Problem id #ES03: Floor Covering.
Given two positive integers N and M, let us consider a rectangular floor with length N and breadth M. Suppose you have an unlimited supply of square tiles of size $2^i \times 2^i$ where $i = 0, 1, 2, \ldots$. Write a C program to compute the minimum number of tiles required to cover the given area.

- Input: `N M` (from command line), $0 < N$, M <100000
- Execution: `./a.out N M`
- Output: `The_minimum_number_of_tiles`
- Error Handling: No need to handle input error

Example:

- Input: 1024 256 Output: 4
- Input: 1025 256 Output: 260
- Input: 3 100 Output: 150

[20]

4. Problem id #ES04: Cycle formation.
Given a set of n words, check if they can be rearranged in cycle $s_1, s_2, \ldots, s_n, s_1$ such that the last letter of s_i is the same as the first letter of s_{i+1} for all i.

For example, if the words are: {“hello”, “world”, “other”, “raw”, “duh”} then the output is “Yes”. They can be rearranged as {“hello”, “other”, “raw”, “world”, “duh”}. For example, if the words are: {“aa”, “ab”, “bc”} then the output is “No”.

- Input: A file `input_ES04.txt`. Each line of the file is a string having maximum length 10.
- Output: Just “Yes” or “No”.

- Hint: Create a directed graph considering each word as a vertex. Then check whether it has a Euler cycle or not. You don't have to store the words inside the vertices. You can just give each word an index and use those indices as corresponding vertices.
- Error Handling: No need to handle input error.

[20]

5. Problem id #ES05: Find Articulation points.
   A vertex in an undirected connected graph is an articulation point (or cut vertex) if removing it (and edges incident to it) disconnects the graph.

   In any network the failure of an Articulation point (vertex) leaves the graph disconnected. Therefore finding them is very important.

   - Input: A file `input_ES05.txt` contains the the Adjacency Matrix of the graph.
   - Output: Print the Articulation vertices (indices).
   - Hint: For a vertex v, remove the vertex v and all the edges incident to it. Then using DFS check the number of connected components. If it is more than one then it is disconnected, and v is a Articulation point. Do it for each vertex.
   - Error Handling: No need to handle input error.

[20]