Introduction to Data Structures

Jinkyu Lee

Dept. of Computer Science and Engineering,
Sungkyunkwan University (SKKU)

Homework 4C

- 10 points for coding evaluation
  - Submission format
    - File name: yourid_HW4C.c
      - Example: 2000123456_HW4C.c
    - File type: .c (NOT .cpp)
  - Submission site: https://icampus.skku.edu
    - Week 11: [Homework] 4C (code)

- No report

- Due date
  - 11/24 23:59 (no late submission accepted)

Rules for homework

- You should follow instructions.
  - Complier
    - You will get no/less point if your program cannot be complied with the specified complier
  - Input/output format
    - You will get no/less point if TA’s automatic evaluation program cannot parse your input or output.
  - Permitted modification scope
    - You will get no/less point if you modify code outside of the permitted modification scope
  - All other rules
    - You will get severe penalty or no/less point if you violate the given rules.

Complier and input/output rules for homework

- Every implementation homework will be evaluated by TA's automatic evaluation program with the following complier.
    - Complier: GCC 7.X, 8.X, 9.X or 10.X
        - https://gcc.gnu.org/
    - You will get no/less point if your program cannot be complied with GCC 7.X, 8.X, 9.X or 10.X.
        - For example, do not rely on visual studio.
    - You can use standard library such as *stdlib.h* and *math.h*.

- Input/output format
    - You will get no/less point if TA's automatic evaluation program cannot parse your input or output according to the following rules.
    - Use `stdin` and `stdout`

Problem

- Problem: Graph to Tree
  - You have the following graph.
    - Your data is either ‘A’, ‘B’, ‘C’, ‘D’, ‘E’, ‘F’, … ‘Z’, meaning that your data is an upper-case letter.
    - ~~For given graph, count the number of each element through graph traversal~~
    - Node id starts from 0 and then 1, 2, 3, ….
    - The given graph is a connected component.
    - The given graph becomes a tree after removing one edge.
  - Find one edge (out of some edges) that makes the graph a tree after removing it.

Input/Ouput

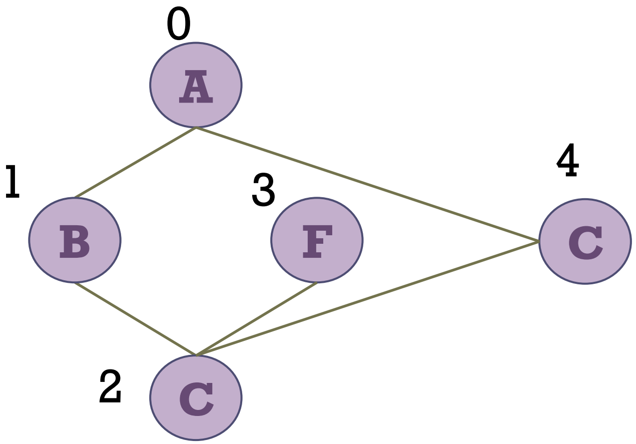
- Input

```
5 5
ABCFC
0 1
0 4
1 2
2 3
2 4
```

# of nodes # of edges

Data of nodes 0, 1, 2, …

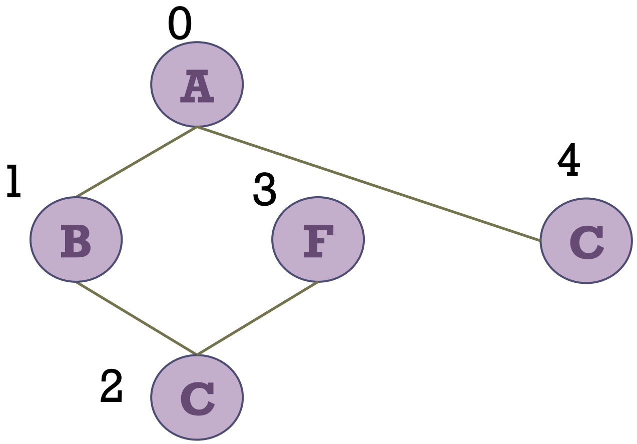
edges



- Output

```
2 4
```

The edge to be removed (if multiple edges are possible, print one of them)



Template

- No template
  - But, you can re-use the template for Homework 4B.

Evaluation

- Evaluation
  - TA will test several cases.
  - For each test case,
    - If your C code results in an answer within 10 seconds on a platform with average computing power,
      - If your output is perfect.
        - You get 100%.
      - Else,
        - You get 0%.
    - Else,
      - You get 0%.