CSC3100 Data Structures Fall 2022

Programming Assignment II

Due: Dec 11 2022

1 Problem 1: Special Shortest Path

1.1 Statement

City C consists of n nodes, representing different places. There are m edges between these nodes. For the edge $e_i = (u_i, v_i, w_i)$, there is a bidirectional(undirected) trail connecting u_i and v_i with length of w_i.

For a path $P = \{p_i\}$, consisting of edges $p_1, p_2, p_3, \cdots, p_k$, the length of each edge is $l_i = w_{p_i}$. Normally, passing the edge p_i with length l_i will cost l_i units of energy. Specially, if $l_i = K \cdot l_{i-1}$, then passing this edge will only cost $(K-1) \cdot l_{i-1}$ units of energy.

Alice is starting from the node 1. Alice wants to know how many units of energy it will take at least to visit the node x, for any x. If x is unreachable from the start point(node 1), you should output -1 as the result.

1.2 Input Format

The first line consists of three integer numbers n, m, K. The following m lines each consists of three integer numbers u, v, w to describe a bidirectional trail.

1.3 Output Format

You need to output a line consisting of n integers, each representing the minimum units of energy to reach node i from node 1.

1.4 Example

Input 1	Output 1
4 3 0 1 2 2 2 3 4 3 1 5	0 2 5 -1

Input 2	Output 2
3 3 2 1 2 2 2 3 4 3 1 5	0 2 4

1.5 Constraints

Case	Score	Constraints			
$1 \sim 3$	30 pts	$n \leq 10^5$	$m \leq 2 \times 10^5$	K = 0	$1 \leq w_i \leq 10^4$
$4 \sim 5$	20 pts	$n \leq 10^3$	$m \leq 2 \times 10^5$	K = 1	
$6 \sim 7$	20 pts	$n \leq 10^5$	$m \leq 2 \times 10^5$	K = 2	
$8 \sim 10$	30 pts	$n \leq 10^5$	$m \leq 2 \times 10^5$	$K \leq 10^5$	

1.6 Hints

- Hint: You can **modify(or add) some edges to the original graph** to fit this problem into the algorithm you know.

2 Problem 2: Median Search Tree

2.1 Statement

Barbara has got a set of values with size of $2k$. Barbara wants to do several types of operations on it. If the sorted array of all the values on the nodes is $\{a_i\}_{i=1}^{n}$, let $t = \lceil n/2 \rceil$, then the median $2k$ values are $\{a_{t-k+1}, \cdots, a_{t+k}\}$.

To make sure that there are at least $2k$ numbers in the set, the initial set contains $2k$ numbers. Then, we need to do m operations belonging to the following 3 types:

- **1 w**: insert a value w.
- **2**: output all the median $2k$ values, i.e. $a_{t-k+p}, \forall 1 \leq p \leq 2k$.
- **3 p**: delete the p-th value among median $2k$ values, i.e. a_{t-k+p}.

We guarantee that all the values will be distinct.

2.2 Input Format

The first line consists of two integer numbers n, k. The second line consists of the $2k$ values in the initial set. Then, the following n lines each consists of the command of an operation.

2.3 Output Format

You need to output one line for each query(operation 2). Each line consists of $2k$ positive integers, the median $2k$ values of the set at that time.

2.4 Example

Input 1	Output 1
3 1 2 3 1 4 3 1 2	2 4

Input 2	Output 2
5 2 8 4 2 6 2 1 5 2 1 3 2	2 4 6 8 4 5 6 8 3 4 5 6

2.5 Constraints

Case	Score	Constraints			
$1 \sim 3$	30 pts	$n \leq 2 \times 10^3$	$k \leq 25$		$1 \leq w \leq 10^6$
$4 \sim 5$	20 pts	$n \leq 10^5$	$k \leq 25$	no operation 3	
$6 \sim 7$	20 pts	$n \leq 10^5$	k = 1		
$8 \sim 10$	30 pts	$n \leq 10^5$	$k \leq 25$		

2.6 Hints

You can solve this problem with **heap**.

3 Problem 3: Football Match

3.1 Statement

While the FIFA World Cup is being held in Qatar, BLGG is organizing a football tournament in LGU, too.

There are n teams in this tournament, numbered from 1 to n. Each team has its popularity, and the popularity of team i is a_i. A match between i and j will gain $a_i \times a_j \ MOD \ M$ attractions.

When a football team loses a match, it will be eliminated from the tournament. At the end, the team left standing will be the champion of this tournament.

BLGG is wondering that what the maximum sum of the attractions of the $(n-1)$ matches.

3.2 Input Format

The first line contains two integers n and M.
The second line contains n integers $a_1, \cdots, a_n$.

3.3 Output Format

Output one integer representing the maximum sum of the attractions of the $(n-1)$ matches.

3.4 Sample Input/Output

Input 1	Output 1
3 114514 1 2 3	9

3.5 Constraints

Case	Score	Constraints		
1	10 pts	$n \leq 10$	$0 \leq a_i, M \leq 2 \times 10^9$	For all $i, j, a_i \times a_j < M$
$2 \sim 5$	40 pts	$n \leq 10$	$0 \leq a_i, M \leq 2 \times 10^9$	
6	10 pts	$n \leq 2000$	$0 \leq a_i, M \leq 2 \times 10^9$	For all $i, j, a_i \times a_j < M$
$7 \sim 10$	40 pts	$n \leq 2000$	$0 \leq a_i, M \leq 2 \times 10^9$	

3.6 Hints

You can try to solve this problem using the graph algorithms we learn in classes.

4 Problem 4: Prefix

4.1 Statement

You are given n strings $s_1, s_2, \cdots, s_n$ and q queries. In i^{th} query, you are given a string t_i, please find out how many strings in $s_1, s_2, \cdots, s_n$ begins with t_i.

4.2 Input Format

The first line is an integer n.
Each of the next n lines contains a string, respectively. The $(i+1)^{th}$ line of input is s_i.
The $(n+2)^{th}$ line of input is an integer q.
Each of the next q lines contains a string, respectively. The $(n+2+i)^{th}$ line of input is t_i.

4.3 Output Format

Output q lines. The i^{th} line contains the answer of i^{th} query.

4.4 Sample Input/Output

Input 1	Output 1
3 wenwen wenbl blgg 3 wen bl csc	2 1 0

4.5 Constraints

All strings only contain lowercase letters.

Case	Score	Constraints	
$1 \sim 4$	40 pts	$n, q \leq 10^3$	$\sum_{i=1}^{n} \lvert s_i \rvert, \sum_{i=1}^{q} \lvert t_i \rvert \leq 10^3$
$5 \sim 10$	60 pts	$n, q \leq 10^6$	$\sum_{i=1}^{n} \lvert s_i \rvert, \sum_{i=1}^{q} \lvert t_i \rvert \leq 10^6$

4.6 Hints

You can try to store $s_1, s_2, \cdots, s_n$ in a tree.
Since the input might be very large, fast input method such as BufferedReader in Java is required.

A. Requirements

Code (90%)

You can write your code in Java, Python, C, or C++. The *time limit* may vary among different languages, depending on the performance of the language. Your code must be a complete runnable program instead of only a function. We guarantee test data strictly compliance with the requirements in the description, and you do not need to deal with cases where the input data is invalid.

We provide a example problem to better illustrate the information above.

Report (10%)

You also need to write a report to explain the following:

- What are the possible solutions for the problem?
- How do you solve this problem?
- Why is your solution better than others?

Please note that the maximum number of pages allowed for your report is **5 pages**.

Remember that the report is to illustrate your thinking process Keep in mind that your report is supposed to show your ideas and thinking process. We expect clear and precise textual descriptions in your report, and we do not recommend that you over-format your report.

B. Example Problem: A + B Problem

Description

Given 2 integers A and B, compute and print A + B

Input

Two integers in one line: A, and B

Output

One integer: A + B

Sample Input I	Sample Output I
1 2	3

Problem Scale & Subtasks

For 100% of the test cases, $0 \leq A, B \leq 10^6$

Solutions

Java

```java
import java.util.*;

public class Example {
    public static void main(String[] args) {
        int a, b;
        Scanner scanner = new Scanner(System.in);
        a = scanner.nextInt();
        b = scanner.nextInt();
        scanner.close();
        System.out.println(a + b);
    }
}
```

Python

```python
AB = input().split()
A, B = int(AB[0]), int(AB[1])
print(A + B)
```

C

```c
#include <stdio.h>

int main(int argc, char *argv[])
{
  int A, B;
  scanf("%d%d", &A, &B);
  printf("%d\n", A + B);
  return 0;
}
```

C++

```cpp
#include <iostream>

int main(int argc, char *argv[])
{
  int A, B;
  std::cin>> A >> B;
  std::cout<< A + B << std::endl;
  return 0;
}
```

C. Submission

After finishing this assignment, you are required to submit your code to the Online Judge System (OJ), and upload your `.zip` package of your code files & report to Black Board.

C.1 Online Judge

Once you have completed one problem, you can submit your code on the page on the Online Judge platform (cuhkszoj.com, campus only) in order to gain marks for the code part. You can submit your solution of one problem for **no more than 30 times**. After you have submitted your program, OJ will test your program on all test cases and give you grade. The grade of your lastest submission will be regarded to as the final grade of the corresponding problem. Each problem are tested on multiple test cases of different difficulty. You will get a part of the score even if your algorithm is not the best.

Note: The program running time may vary on different machines, please refer to the result on the online judge system. OJ will show the time and memory limits for different languages on the corresponding problem page.

OJ access code: `CSC3100assignment2`, you are using this code whenever you are asked to do so.

If you have other questions about the online judge system, please refer to OJ wiki (campus network only). And if this cannot help you well, feel free to contact us.

C.2 BlackBoard

You are required to upload your **source codes and report** to the BlackBoard platform. You need to name your files according to the following rules and compress them into `A2_<Student ID>.zip` :

```
A2_<Student ID>.zip:
    A2_P1_<Student ID>.java/py/c/cpp/cc
    A2_P2_<Student ID>.java/py/c/cpp/cc
    A2_P3_<Student ID>.java/py/c/cpp/cc
    A2_P4_<Student ID>.java/py/c/cpp/cc
    A2_Report_<Student ID>.pdf
```

For example, suppose your ID is 123456789, and your problem 1 is written in `Python`, problem 2 is written in `Java`, problem 3 is written in `C` and problem 4 in `C++`, then the following contents should be included in your submitted `A2_123456789.zip`:

```
A2_P1_123456789.py
A2_P2_123456789.java
A2_P3_123456789.c
A2_P4_123456789.cpp
A2_Report_123456789.pdf
```

Note

If you have questions for the problems above, please contact:

- Problem 1 & 2. Yuan Xu: yuanxu1@link.cuhk.edu.cn
- Problem 3 & 4. Biaolin Wen: biaolinwen@link.cuhk.edu.cn
- OJ issues: Tianci Hou: 121090184@link.cuhk.edu.cn