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 excutable 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.

Libraries in this assignment:

- For C/C++, you can only include standard library.
- For Java, you can only `import java.util.*`
- For Python, you can only import standard library. In other words, you cannot import libraries such as `numpy`.

Report (10%)

You also need to write a report in `pdf` type 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 1	Sample Output 1
`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 and report to BlackBoard.

C.1 Online Judge

Once you have completed one problem, you can submit your code on the page on the Online Judge platform (oj.cuhk.edu.cn, campus only) to gain marks for the code part. You can submit your solution of one problem for **no more than 80 times**.

After you have submitted your program, OJ will test your program on all test cases and give you a grade. The grade of your latest submission will be regarded as the final grade of the corresponding problem. Each problem is 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 of the online judge system. OJ will show the time and memory limits for different languages on the corresponding problem page.

If you have other questions about the online judge system, please refer to OJ wiki (campus network only). If this cannot help you, 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 `A1_<Student ID>.zip` :

```
A1_<Student ID>.zip
|-- A1_P1_<Student ID>.java/py/c/cpp
|-- A1_P2_<Student ID>.java/py/c/cpp
|-- A1_Report_<Student ID>.pdf
```

For Java users, **you don't need to consider the consistency of class name and file name**.

For example, suppose your ID is 123456789, and your problem 1 is written in `Python`, problem 2 is written in `Java` then the following contents should be included in your submitted `A1_123456789.zip`:

```
A1_123456789.zip
|-- A1_P1_123456789.py
|-- A1_P2_123456789.java
|-- A1_Report_123456789.pdf
```

C.3 Late Submissions

Submissions after Mar 27, at 23:59 PM (UTC+8) would be considered as LATE. A late submission contest will open after deadline.

Submisson time = max{latest submisson time for every problem, BlackBoard submisson time}

There will be penalties for late submission:

- 0–24 hours after deadline: final score = your score×0.8
- 24–72 hours after deadline: final score = your score×0.5
- 72+ hours after deadline: final score = your score×0

FAQs

Q: I cannot access to Online Judge.

A: First, please ensure that you are using the campus network. If you are not on campus, please use the university VPN. Second, please delete cookies and refresh browser or use other browser. If you still cannot access to Online Judge, try to visit it via the IP address `10.26.200.13`.

Q: My program passes samples on my computer, but not get AC on OJ.

A: Refer to OJ Wiki Q&A

Authors

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

- Problem 1. The 2024 Programming Contest of CUHK-Shenzhen (official)
- Problem 2. Yingli Zhou: yinglizhou@link.cuhk.edu.cn Ziyi Zhao: ziyizhao2@link.cuhk.edu.cn

CSC3100 Data Structures Spring 2024

Programming Assignment 3

Due: May 5 2024 23:59:00

Assignment Link: http://oj.cuhk.edu.cn/contest/csc310024spa4

Access code: flute

Question 1 weighs 40% and question 2 weighs 50%.

Please note that you are not recommended to use AI tools such as chatGPT to complete your assignment for potential plagiarism issue.

1 Is A DFS Order (40% of this assignment)

1.1 Description

Given a tree T rooted at vertex 1 and a permutation A of the set $[N] = \{1, 2, \ldots, N\}$, the task is to determine whether A can be a valid Depth-First Search (DFS) order of T.

Note that a tree can have more than one valid DFS ordering, as the order of children for each node is not predetermined.

1.2 Input

The first line of the input contains an integer T, representing the number of test cases. Each test case is given as follows:

- The first line contains an integer $N (N \geq 1)$, denoting the number of vertices in the tree.
- The second line contains $N-1$ integers $p_2, p_3, \ldots, p_N (1 \leq p_i \leq i-1)$, where p_i denotes the parent of vertex i in T.
- The third line contains a permutation A of $[N]$. It is guaranteed that the first element of A is 1.

Algorithm 1 DFS on a rooted tree

```
1:  procedure DFS(root)
2:      visited ← empty set
3:      DFS-Visit(root, visited)
4:  end procedure
5:
6:  procedure DFS-Visit(node, visited)
7:      add node to visited
8:      for all child of node do
9:          if child is not in visited then
10:             DFS-Visit(child, visited)
11:         end if
12:     end for
13: end procedure
```

Figure 1: The pseudo-code of DFS

1.3 Output

For each test case, output a single line containing either "YES" or "NO", indicating whether the given permutation can be a valid DFS order of T.

Sample Input 1

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

Sample Output 1

```
YES
NO
```

You can find more samples in the attached file on BB.

1.4 Problem Scale & Subtasks

There are 6 testcases in total. It is guaranteed that the sum of N over all test cases does not exceed 10^5.

2 Gift (50% of this assignment)

2.1 Description

Piggy wants to buy some gifts for his friends. Specifically, he wants to buy N gifts. Each of them sells for $\$W$.

Now the gift shop has a sale event. If you buy gift i, you only need to pay more $\$A_{ij}$ to get gift j if $A_{ij} \neq 0$. For all i, j, $A_{ij} = A_{ji}$.

Piggy wants to know the least money he needs to pay. You need to help him to find the answer.

2.2 Input

The first line contains two integers N, W.

The following N lines, each line with N integers. The j_{th} integer in the i_{th} line is A_{ij}. It is guaranteed that $A_{ij} = A_{ji}$ and $A_{ii} = 0$.

2.3 Output

Output the least money Piggy needs to pay.

Sample Input 1

```
3 3
0 2 4
2 0 2
4 2 0
```

Sample Output 1

```
7
```

You can find more samples in the attached file on BB.

2.4 Problem Scale & Subtasks

$0 \leq A_{ij}, W \leq 1000$

Test Data Scale.	Constraints
30% data	$1 \leq N \leq 10$
100 % data	$1 \leq N \leq 500$

Hint

Try to transform the problem to a spanning tree problem.