Project 5: Saving the World!

CMPE 250, Data Structures and Algorithms, Fall 2020

Instructor: H. B. Yılmaz
TAs: Umut Kocasarı, Rıza Özçelik
Contacts: umut.kocasari@boun.edu.tr
riza.ozcelik@boun.edu.tr

Due: **February 10, Wednesday, 23:55**

1 Introduction

Year 2045... Humanity has gone through lots of global disasters: epidemics, wars, famines, climate change... Now, the world is different from the one humans remember. The Earth is becoming a more and more uninhabitable place for the human-kind, as the scientists declare. But, if there is science, there is always hope! Always!

The visionary scientists had already started to seek solutions to save humanity and discovered a new species in space. This new species has been the dominant species of its planet for hundreds of millions of years, whereas we, the human-kind, are about to become extinct only in two million years. The scientists investigated the reason behind this and found a simple, yet striking answer: altruism.

The members of this species are all altruistic and care about each other as much as they care about themselves. They behave as a single unit and seek a global maximum for the community, as opposed to seeking self-maximums, which inevitably cause global disasters. Then the question arises: why did they evolve so differently from the human-kind?

The research revealed surprising answers. Similar to human-kind, the members of this species were also too self-oriented initially and cared about only themselves and their close relatives. This resulted in global "disasters" which eventually made them lose the ability to recognize their loved ones, except for their parents. First, they could not know how to treat others as they could not tell if an individual was a part of their family. But then, they have come up with an excellent idea: if we cannot recognize our close ones, why not treat everyone as if they are family?

They internalized this notion so much that they transformed this into an altruistic culture of sharing and caring, and built a sustainable world. They still would love to remember their family members though, and this may be our way to establish communication with them!

We know that each member knows only its parent and would love to learn its greater ancestors. Since there are a lot of individuals in this society, they need an efficient way to find out their ancestors. Luckily, we can design an algorithm to help them, and then hope that, they would teach us to secrets to happy and long cohabitation...

The scientists gathered further information about the species that we can utilize to tackle the problem.

- The individuals in this society have exactly one parent, except for the first individual that has no parent, the root.
- Each individual derives from the same root.
- The individuals might have an arbitrary number of children.
- Each individual remembers only its parent and does not know further information about its family. The root knows that it has no parent.

We learned all parent-child relations and now they are dying to learn who their n^{th}ancestors are. Thus, we need an efficient way to answer their queries. Luckily, we are aware that this problem is a slightly different version of the Level Ancestor problem and know an algorithm[1] for the solution. The world desperately needs you to implement this algorithm... Would you help us? Would you save the world?

2 Input & Output

Your code must read the name of the input and output files from the command line. We will run your code as follows:

```
g++ *.cpp *.h -std=c++11 -o project5.out
./project5.out inputFile outputFile
```

If you do not use any ".h" file. Your code will be compiled as follows: `g++ *.cpp -std=c++11 -o project5.out`

Make sure that your final submission compiles and runs with these commands

2.1 Input

The first line of the input file states the number of individuals I ($1 \leq I \leq 1000000$) and the second line contains the parent id of each individual. Here, the id of an individual is equal to its zero-based index in this line and the root has the parent -1. The third line comprises the number of queries Q and is followed by Q lines ($1 \leq Q \leq 500000$), in which the individuals ask who their n^{th} ancestors are. Each query line contains two integers: the id of the individual and n, separated by a space. The input is guaranteed to form a tree of parent-child relations. Below is the input template; Table 1 displays an example input file on the left and Figure 1 illustrates the example input as a tree.

[1]Here is a link to a simple algorithm: `https://arxiv.org/abs/1903.01387`. The manuscript contains a small mistake in its Section 2.3 though. The second item of this section should start with "Look up the largest element `s smaller` than `l`"

Input	Output
9	0
-1 0 1 0 3 2 4 1 0	3
4	0
1 1	-1
4 1	
4 2	
6 4	

Table 1: Sample input and output files

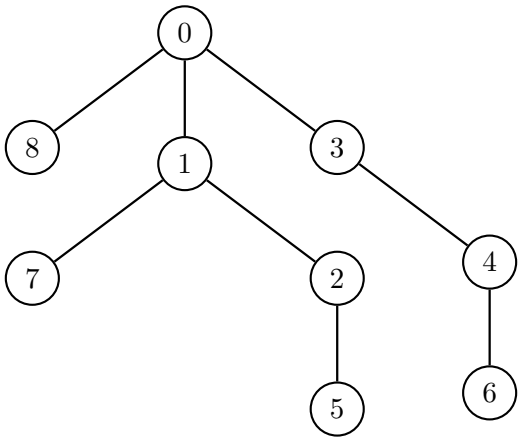

Figure 1: Tree representation of the the sample input file.

I
`parent_id_of_id_0 parent_id_of_id_1 parent_id_of_id_2 ... parent_of_id_`$I-1$
Q
`query1`
`query2`
`query3`
⋮
`query`Q

2.2 Output

The output file must consist of Q lines where each line contains the answer to the corresponding query. If the query is invalid (i.e. such an ancestor does not exist) print -1. Table 1 displays the output file for the sample input on the right and Table 2 explains each answer.

Query-1 (1 1)	The query asks the first ancestor (parent) of the individual 1. The tree in Figure 1 displays that the first ancestor of the individual 1 is the individual 0, and thus the answer is also 0.
Query-2 (4 1)	The query asks the first ancestor of the individual 4, which is the individual 3.
Query-3 (4 2)	The query asks the second ancestor (parent of its parent, or grand-parent) of the individual 4. Figure 1 shows that the parent of the individual 4 is the individual 3 and its parent is the individual 0. Thus, the answer is also 0.
Query-4 (6 4)	The query asks fourth ancestor of the individual 6, which does not exist. Thus the answer is -1.

Table 2: Explanations for the expected output file.

3 Grading

This is an optional project that would replace your lowest project grade **if it is in the range [0, 65]**. In other words, **if you are satisfied with your grades for the previous projects or they are all higher than 65, you do not have to submit any code**. Additionally, **the maximum grade for this project is set to 75**, instead of 100, for fairness.

We will automatically grade your submission with multiple test cases besides the ones we already provided. So, **ensure that your submission is runnable with the commands we provided above and satisfies all project requirements.** The auto-runnability of the submission will constitute **5/75** of your final grade and passing the test cases will earn you **70/75** more points. If your submission is not auto-runnable, then you will receive 0 at first and then have to issue an objection.

Notes and Constraints

- There will be time limit on test cases, so it is important to write your codes in an efficient manner.
- Passing all given test cases does not necessarily mean that your submission will pass the grading test cases. So, make sure to write your code correctly and it extrapolates to new test cases.
- You are allowed to use the standard C++ library, but not additional libraries. All requirements of the project must be satisfied with your authentic code, not by any C++ library. **Otherwise, you will get zero**.

Submission Details

You are supposed to use GitHub Classroom for the submission. No other type of submission will be accepted. Also, pay attention to the following points:

- All source codes are checked automatically for similarity with other submissions and exercises from previous years. **Make sure you write and submit your own code. Any sign of cheating will be penalized by at least -100 points.**

- You can add as many files as you can as long as they are in the same folder with main.cpp.
- Make sure you document your code with necessary inline comments and use meaningful variable names. Do not over-comment, or make your variable names unnecessarily long. This is very important for partial grading.