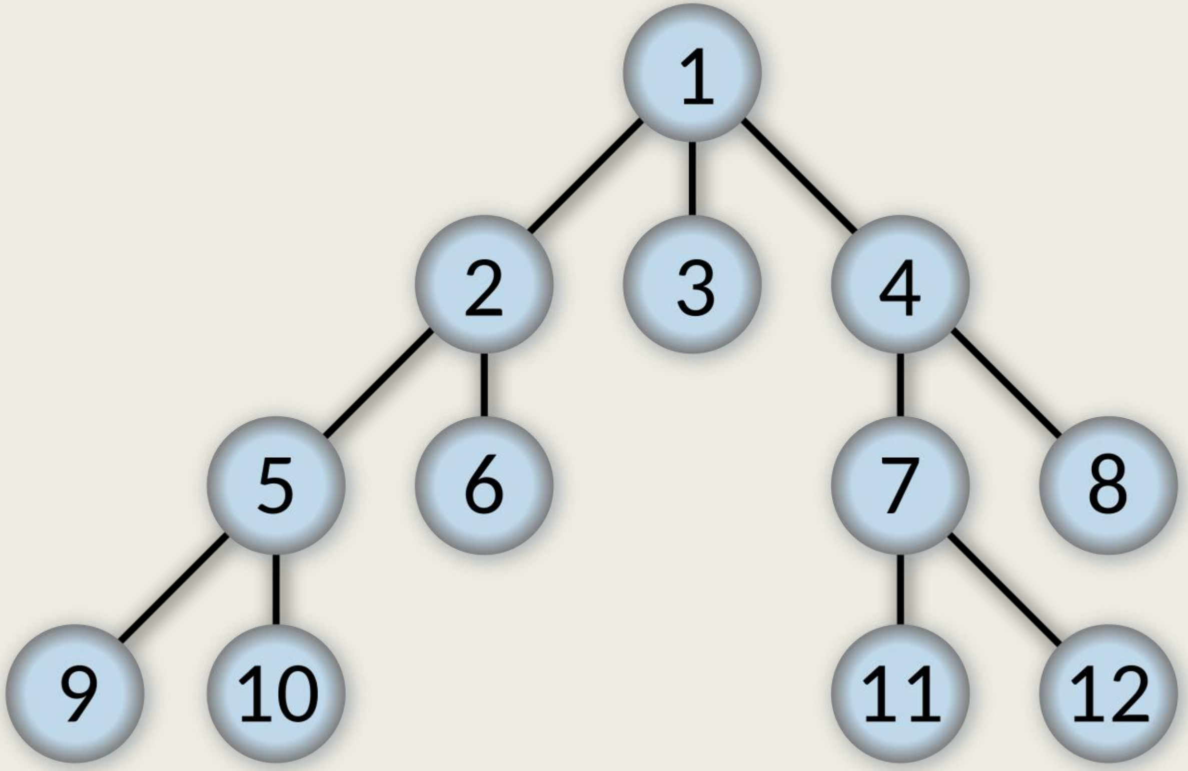
CCC JUNIOR LEC10

Topic: BFS (Breadth First Search) & DFS (Depth First Search)

今日课程预览

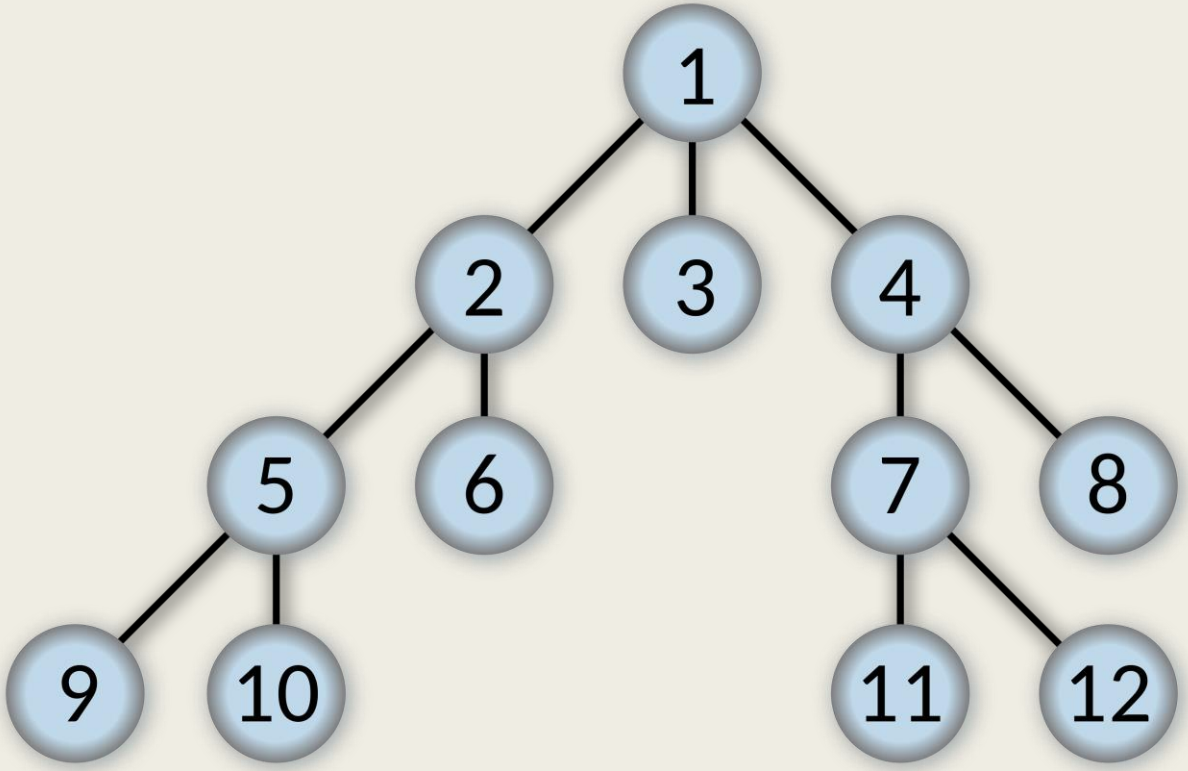
- Review of last week’s problem (coin problem, pie-day, arrival time).
- BFS
- DFS
- Past exams

BFS



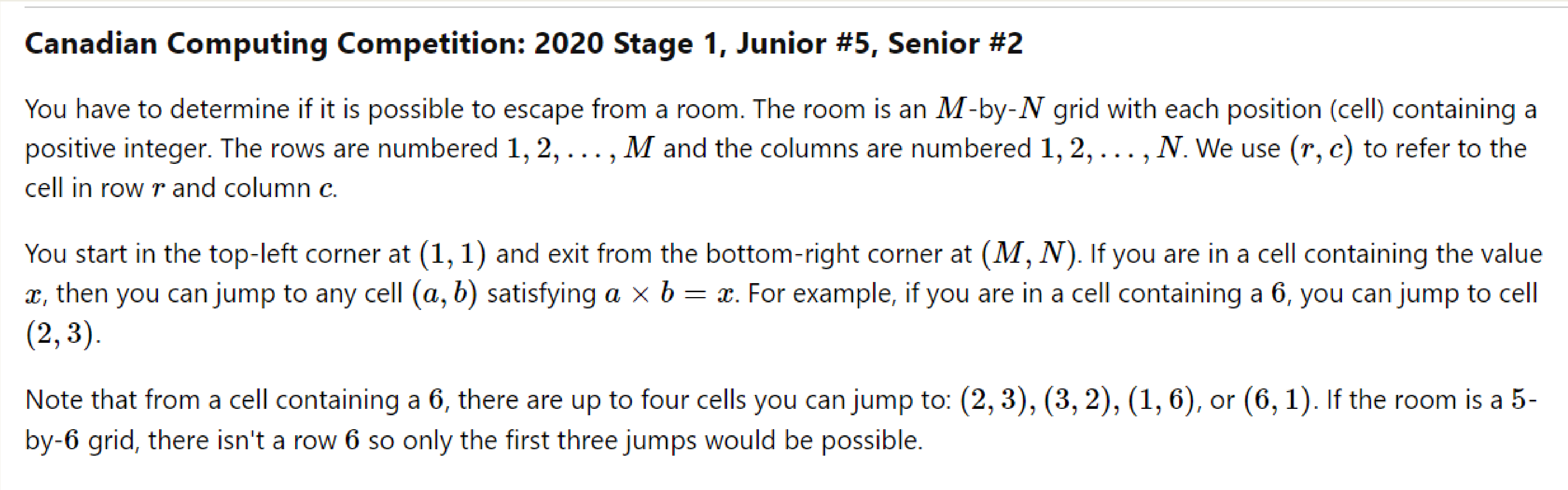
Dynamic queue in BFS:

DFS



Dynamic stack in DFS:

Example: Past Exam

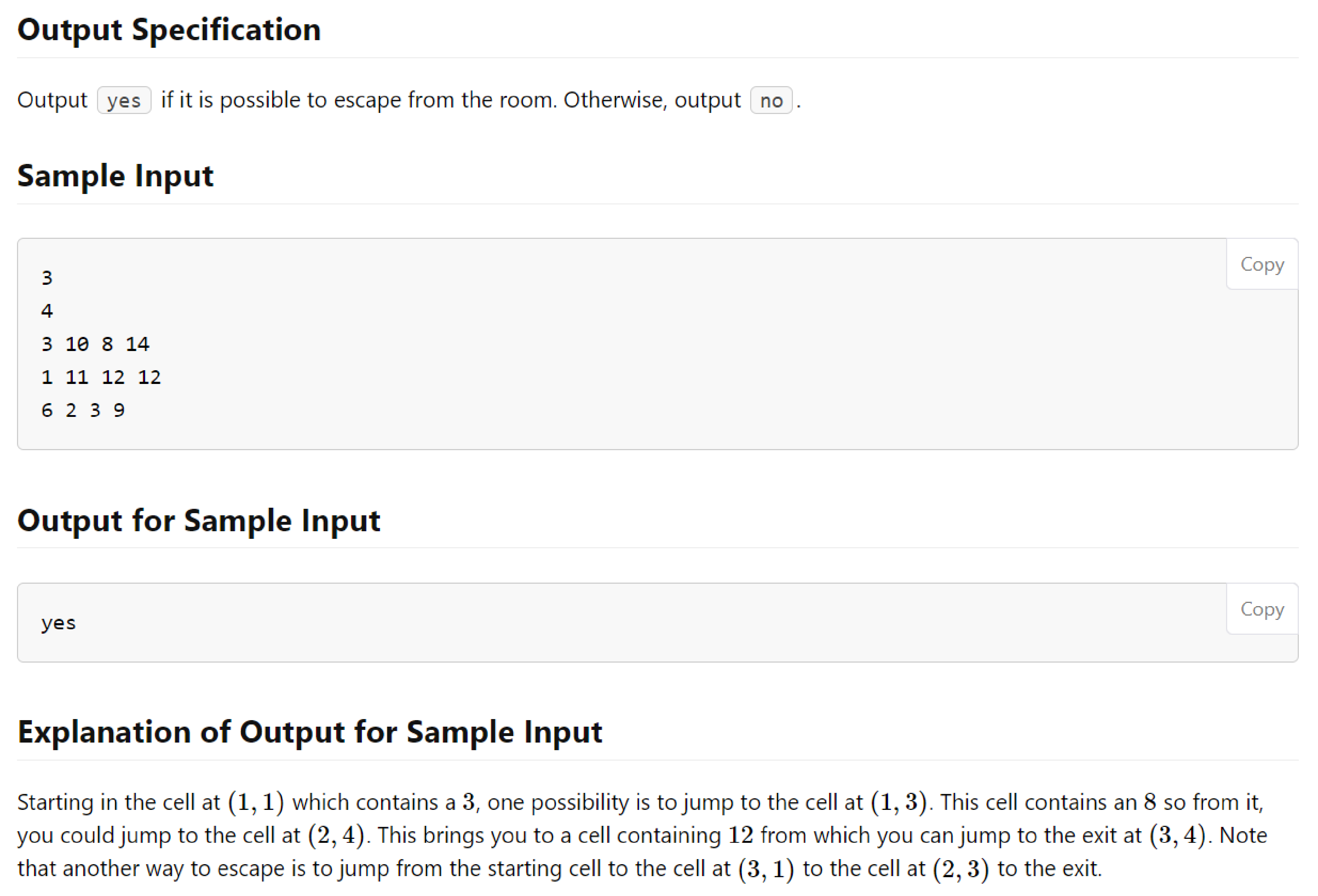


Canadian Computing Competition: 2020 Stage 1, Junior #5, Senior #2

You have to determine if it is possible to escape from a room. The room is an M-by-N grid with each position (cell) containing a positive integer. The rows are numbered $1, 2, \ldots, M$ and the columns are numbered $1, 2, \ldots, N$. We use (r, c) to refer to the cell in row r and column c.

You start in the top-left corner at $(1, 1)$ and exit from the bottom-right corner at (M, N). If you are in a cell containing the value x, then you can jump to any cell (a, b) satisfying $a \times b = x$. For example, if you are in a cell containing a 6, you can jump to cell $(2, 3)$.

Note that from a cell containing a 6, there are up to four cells you can jump to: $(2, 3)$, $(3, 2)$, $(1, 6)$, or $(6, 1)$. If the room is a 5-by-6 grid, there isn't a row 6 so only the first three jumps would be possible.



Output Specification

Output `yes` if it is possible to escape from the room. Otherwise, output `no`.

Sample Input

```
3
4
3 10 8 14
1 11 12 12
6 2 3 9
```

Output for Sample Input

```
yes
```

Explanation of Output for Sample Input

Starting in the cell at $(1, 1)$ which contains a 3, one possibility is to jump to the cell at $(1, 3)$. This cell contains an 8 so from it, you could jump to the cell at $(2, 4)$. This brings you to a cell containing 12 from which you can jump to the exit at $(3, 4)$. Note that another way to escape is to jump from the starting cell to the cell at $(3, 1)$ to the cell at $(2, 3)$ to the exit.

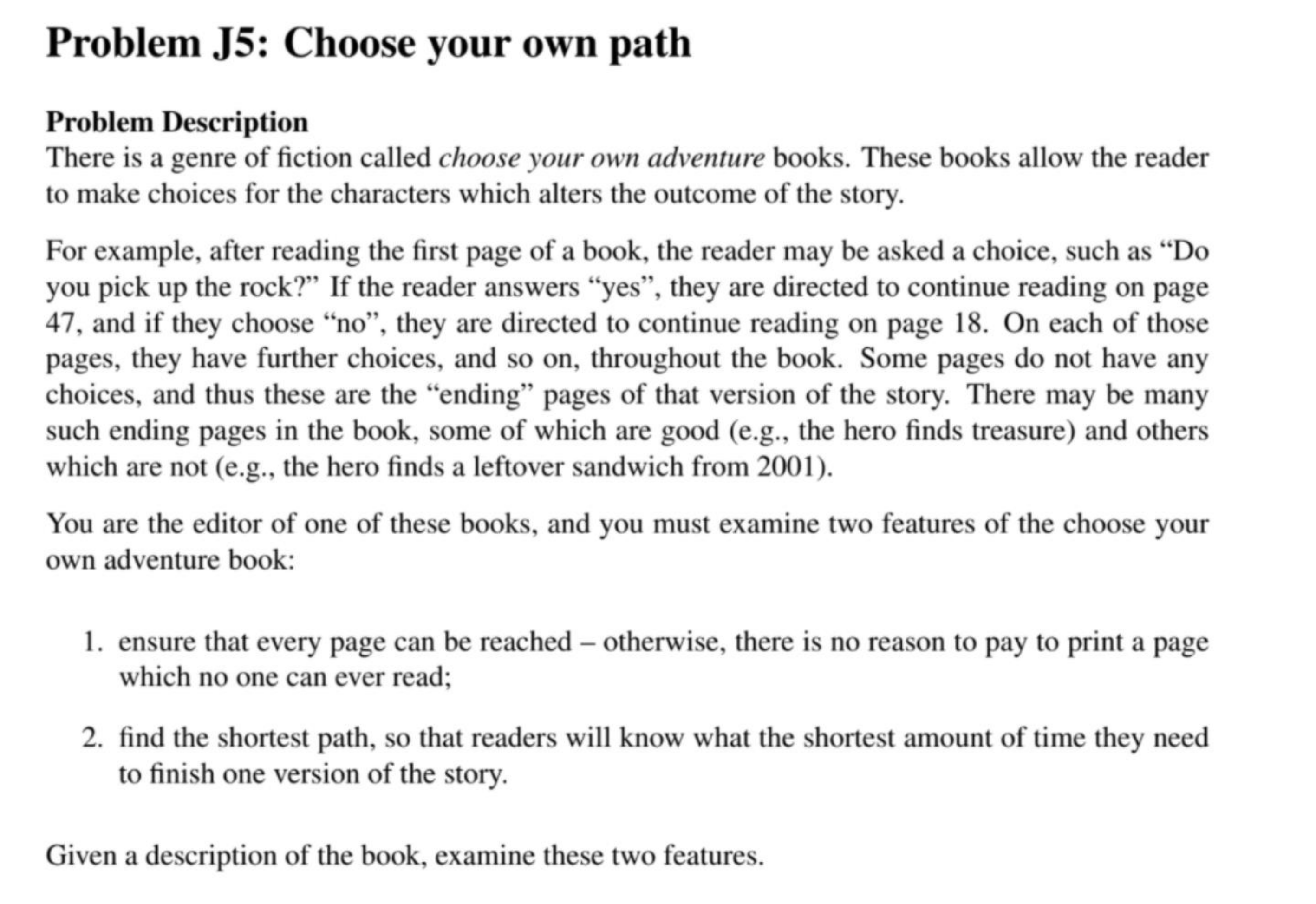
Solve the problem in BFS

- What data structure should be used as list to contain the status for checking?
- What is the pop up mechanism for this list.
- What data structure should be used to determine visited slots/status?
- Can we improve the program to let it be faster?

Solve the problem in DFS

- What data structure should be used as list to contain the status for checking?
- What is the pop up mechanism for this list?
- What data structure should be used to determine visited slots/status?

Another example: PAST EXAM



Problem J5: Choose your own path

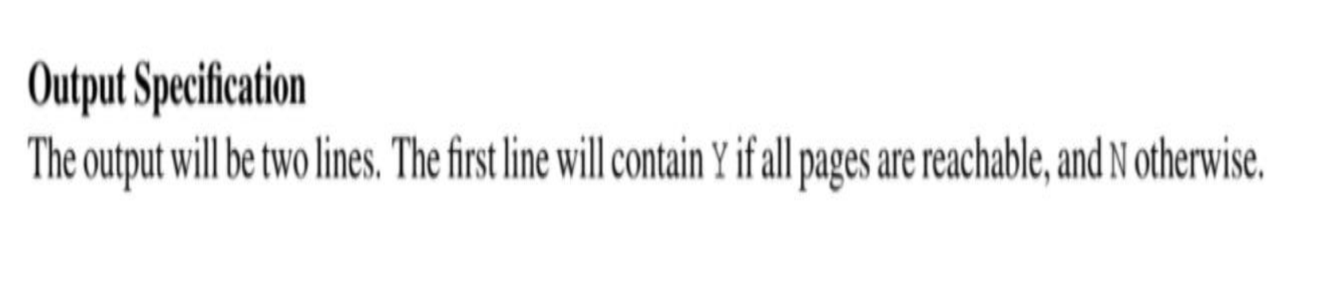
Problem Description
There is a genre of fiction called *choose your own adventure* books. These books allow the reader to make choices for the characters which alters the outcome of the story.

For example, after reading the first page of a book, the reader may be asked a choice, such as "Do you pick up the rock?" If the reader answers "yes", they are directed to continue reading on page 47, and if they choose "no", they are directed to continue reading on page 18. On each of those pages, they have further choices, and so on, throughout the book. Some pages do not have any choices, and thus these are the "ending" pages of that version of the story. There may be many such ending pages in the book, some of which are good (e.g., the hero finds treasure) and others which are not (e.g., the hero finds a leftover sandwich from 2001).

You are the editor of one of these books, and you must examine two features of the choose your own adventure book:

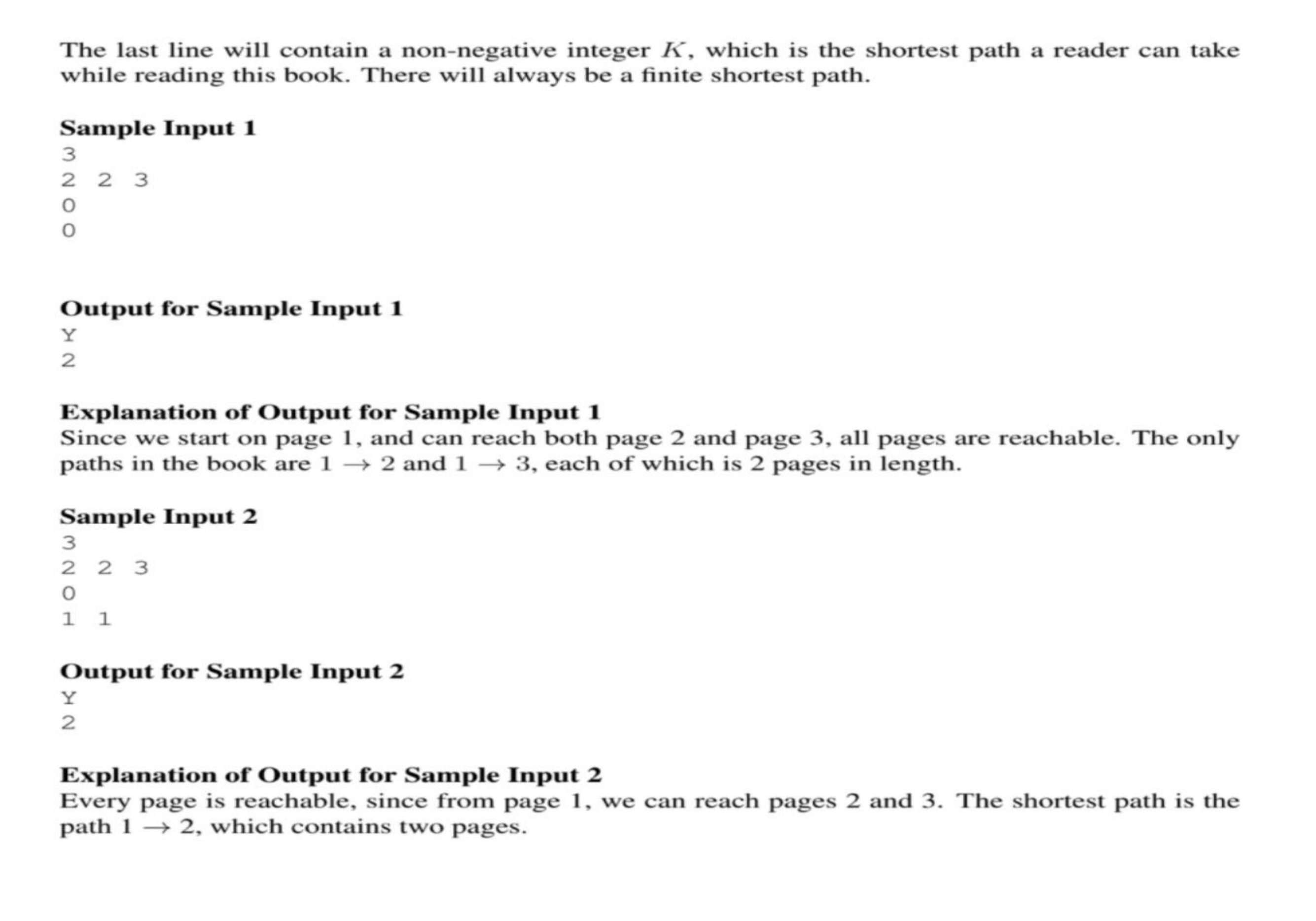
1. ensure that every page can be reached – otherwise, there is no reason to pay to print a page which no one can ever read;
2. find the shortest path, so that readers will know what the shortest amount of time they need to finish one version of the story.

Given a description of the book, examine these two features.



Output Specification

The output will be two lines. The first line will contain Y if all pages are reachable, and N otherwise.



The last line will contain a non-negative integer K, which is the shortest path a reader can take while reading this book. There will always be a finite shortest path.

Sample Input 1

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

Output for Sample Input 1

```
Y
2
```

Explanation of Output for Sample Input 1

Since we start on page 1, and can reach both page 2 and page 3, all pages are reachable. The only paths in the book are $1 \rightarrow 2$ and $1 \rightarrow 3$, each of which is 2 pages in length.

Sample Input 2

```
3
2 2 3
0
1 1
```

Output for Sample Input 2

```
Y
2
```

Explanation of Output for Sample Input 2

Every page is reachable, since from page 1, we can reach pages 2 and 3. The shortest path is the path $1 \rightarrow 2$, which contains two pages.