Assignment 3

CS1.201: Data Structures and Algorithms

Hash Tables

Due Date : 2nd March 2024
Total Marks: 100

1 General Instructions

- The assignment must be implemented in C.
- The assignment needs to be submitted on OJ. This assignment comprises one question with two subtasks. Note that there will be two portals on OJ, one for each subtask. Even if you have submitted the second subtask, you must submit both subtasks separately to obtain full marks.
- The deadline is strict and will not be extended.
- Resorting to any sort of plagiarism (as mentioned in the policy shared) would be heavily penalised.

2 The Grand Story Conundrum

Kidyansh Akaran, has retired from Belicity Coordinator and has been appointed as a librarian at the Great Library of IIT-H was faced with a task that was deemed impossible for a long time. Each book in that library was represented by a number (which might not be unique). These books were kept in a particular order at all times and could not be moved to any other location. The problem was to find the longest subsequence of books that had

successive numbers on them. This would form a great story made of multiple short stories.

A subsequence of a given sequence is a sequence that can be derived from the given sequence by deleting some or no elements without changing the order of the remaining elements. For example, the sequence ⟨A , B , D⟩ is a subsequence of ⟨A , B , C , D , E , F⟩ obtained after removal of elements C , E and F.

You are given a number `n` and an array `book` representing the numbers on each book. You are supposed to find the longest subsequence in each array where the subsequence is of form `[x, x+1, ..., x+m-1]` where x is any number and m is the length of the subsequence. You have to print the length of such subsequence and the index of occurrence of the elements of the subsequence. (In parituclar if x occurs multiple times before x+1, you may chose any index of x). If there are multiple correct answers, answer any of them.

2.1 I/O Format

Input Format

```
n
book[0] book[1] ...  book[n-1]
```

The first line of the input contains the number of books in the library. The second line contains the number representing each book.

Output Format

```
m
idx[0] idx[1] ...  idx[m-1]
```

The first line of the output contains the length of the longest subsequence. The next line should contain the indices of the elements which are part of the longest subsequence. Each index should be 0-indexed.

2.2 Constraints

- $1 \leq n \leq 2 \cdot 10^5$

- $1 \leq book[i] \leq 10^9$

2.3 Sample Testcases

Input 1

```
9
1 5 6 7 2 8 3 4 10
```

Output 1

```
4
0 4 6 7
```

Explanation 1

The longest subsequence containing continuous numbers is (1 2 3 4). There is another answer of numbers (5,6,7,8) at indexes (1,2,3,5).

Input 2

```
6
1 1 2 3 7 8
```

Output 2

```
3
0 2 3
```

Explanation 2

The longest subsequence containing continuous numbers is (1 2 3). There is another duplicate correct answer due to the duplicate 1.

Input 3

```
8
12 14 15 13 98 14 15 97
```

Output 3

```
4
0 3 5 6
```

Explanation 3

The longest subsequence containing continuous numbers is (12 13 14 15). Note 14,15 at indexes 1,2 respectively can NOT form the longest subsequence. There is only one correct answer for this input.

3 Hints

You can break this problem into two parts. First find any subsequence of maximum length. Then think of tracking its indexes.
Though this may look similar to the longest common subsequence with 1,2...n, it is not the same problem.
Hash tables might be needed with some other concepts you've learned in CPro/DSA to solve this problem :)