ENSF 594

Midterm Project

Due Time: 7th August 2020

What to upload:

A pdf file to describe your code.

Your Code

Money Change Again

As we already know, a natural greedy strategy for the change problem does not work correctly for any set of denominations. For example, if the available denominations are 1, 3, and 4, the greedy algorithm will change 6 cents using three coins $(4 + 1 + 1)$ while it can be changed using just two coins $(3 + 3)$. Your goal now is to apply dynamic programming for solving the Money Change Problem for denominations 1, 3, and 4.

Problem Description

Input Format. Integer *money*.

Output Format. The minimum number of coins with denominations 1, 3, 4 that changes *money*.

Constraints. $1 \leq money \leq 10^3$.

Sample 1.

Input:

```
2
```

Output:

```
2
```

$2 = 1 + 1$.

Sample 2.

Input:

```
34
```

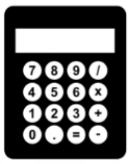
Output:

```
9
```

$34 = 3 + 3 + 4 + 4 + 4 + 4 + 4 + 4 + 4$.

Primitive Calculator

Problem Introduction



You are given a primitive calculator that can perform the following three operations with the current number x: multiply x by 2, multiply x by 3, or add 1 to x. Your goal is given a positive integer n, find the minimum number of operations needed to obtain the number n starting from the number 1.

Problem Description

Task. Given an integer n, compute the minimum number of operations needed to obtain the number n starting from the number 1.

Input Format. The input consists of a single integer $1 \leq n \leq 10^6$.

Output Format. In the first line, output the minimum number k of operations needed to get n from 1. In the second line output a sequence of intermediate numbers. That is, the second line should contain positive integers $a_0, a_2, \ldots, a_{k-1}$ such that $a_0 = 1$, $a_{k-1} = n$ and for all $0 \leq i < k - 1$, a_{i+1} is equal to either $a_i + 1$, $2a_i$, or $3a_i$. If there are many such sequences, output any one of them.

Sample 1.

Input:

```
1
```

Output:

```
0
1
```

Sample 2.

Input:

```
5
```

Output:

```
3
1 2 4 5
```

Here, we first multiply 1 by 2 two times and then add 1. Another possibility is to first multiply by 3 and then add 1 two times. Hence "1 3 4 5" is also a valid output in this case.

Sample 3.

Input:

```
96234
```

Output:

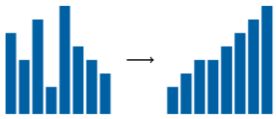
```
14
1 3 9 10 11 22 66 198 594 1782 5346 16038 16039 32078 96234
```

Again, another valid output in this case is "1 3 9 10 11 33 99 297 891 2673 8019 16038 16039 48117 96234".

Improving Quick Sort

Problem Introduction

The goal in this problem is to redesign a given implementation of the randomized quick sort algorithm so that it works fast even on sequences containing many equal elements.



Problem Description

Task. To force the given implementation of the quick sort algorithm to efficiently process sequences with few unique elements, your goal is replace a 2-way partition with a 3-way partition. That is, your new partition procedure should partition the array into three parts: $< x$ part, $= x$ part, and $> x$ part.

Input Format. The first line of the input contains an integer n. The next line contains a sequence of n integers $a_0, a_1, \ldots, a_{n-1}$.

Constraints. $1 \leq n \leq 10^5$; $1 \leq a_i \leq 10^9$ for all $0 \leq i < n$.

Output Format. Output this sequence sorted in non-decreasing order.

Sample 1.

Input:

```
5
2 3 9 2 2
```

Output:

```
2 2 2 3 9
```

Collecting Signatures

Problem Introduction

You are responsible for collecting signatures from all tenants of a certain building. For each tenant, you know a period of time when he or she is at home. You would like to collect all signatures by visiting the building as few times as possible.

The mathematical model for this problem is the following. You are given a set of segments on a line and your goal is to mark as few points on a line as possible so that each segment contains at least one marked point.

Problem Description

Task. Given a set of n segments $\{[a_0, b_0], [a_1, b_1], \ldots, [a_{n-1}, b_{n-1}]\}$ with integer coordinates on a line, find the minimum number m of points such that each segment contains at least one point. That is, find a set of integers X of the minimum size such that for any segment $[a_i, b_i]$ there is a point $x \in X$ such that $a_i \leq x \leq b_i$.

Input Format. The first line of the input contains the number n of segments. Each of the following n lines contains two integers a_i and b_i (separated by a space) defining the coordinates of endpoints of the i-th segment.

Constraints. $1 \leq n \leq 100$; $0 \leq a_i \leq b_i \leq 10^9$ for all $0 \leq i < n$.

Output Format. Output the minimum number m of points on the first line and the integer coordinates of m points (separated by spaces) on the second line. You can output the points in any order. If there are many such sets of points, you can output any set. (It is not difficult to see that there always exist a set of points of the minimum size such that all the coordinates of the points are integers.)

Sample 1.

Input:

```
3
1 3
2 5
3 6
```

Output:

```
1
3
```

In this sample, we have three segments: $[1, 3], [2, 5], [3, 6]$ (of length $2, 3, 3$ respectively). All of them contain the point with coordinate 3: $1 \leq 3 \leq 3$, $2 \leq 3 \leq 5$, $3 \leq 3 \leq 6$.

Sample 2.

Input:

```
4
4 7
1 3
2 5
5 6
```

Output:

```
2
3 6
```

The second and the third segments contain the point with coordinate 3 while the first and the fourth segments contain the point with coordinate 6. All the four segments cannot be covered by a single point, since the segments $[1, 3]$ and $[5, 6]$ are disjoint.