

OJ 10656

Maximum Sum (II)

Prof. Edson Alves – UnB/FGA

Problema

In a given sequence of non-negative integers you will have to find such a sub-sequence in it whose summation is maximum.

Input

The input file contains several input sets. The description of each set is given below:

Each set starts with an integer N ($N < 1000$) that indicates how many numbers are in that set. Each of the next N lines contains a single non-negative integer. All these numbers are less than 10000.

Input is terminated by a set where $N = 0$. This set should not be processed.

Output

For each set of input produce one line of output. This line contains one or more integers which are taken from the input sequence and whose summation is maximum. If there is more than one such subsequence print the one that has minimum length. If there is more than one sub-sequence of minimum length, output the one that occurs first in the given sequence of numbers. A valid sub-sequence must have a single number in it. Two consecutive numbers in the output are separated by a single space.

Exemplo de entradas e saídas

Sample Input

2
3
4
0

Sample Output

3 4

Solução com complexidade $O(N)$

- Como a sequência é composta apenas por números não-negativos, cada elemento x da sequência ou amplia a soma (caso $x > 0$) ou a mantém (caso $x = 0$)
- Assim, a menor sequência máxima é obtida retirando os elementos iguais a zero da sequência
- Importante notar que os elementos devem ser impressos na mesma ordem da entrada
- Há um *corner case* a ser tratado: caso a sequência seja inteiramente composta por elementos iguais a zero, a saída deve ser um único zero, já que o problema estabelece que uma saída válida deve ter no mínimo um número
- Como cada elemento é avaliado uma única vez, a complexidade desta solução é $O(N)$

Solução com complexidade $O(N)$

```
1 #include <bits/stdc++.h>
2
3 using namespace std;
4
5 vector<int> solve(const vector<int>& xs)
6 {
7     vector<int> ans;
8
9     for (auto x : xs)
10         if (x > 0)
11             ans.push_back(x);
12
13     return ans;
14 }
15
16 int main()
17 {
18     ios::sync_with_stdio(false);
19
20     int N;
```

Solução com complexidade $O(N)$

```
22  while (cin >> N, N)
23  {
24      vector<int> xs(N);
25
26      for (int i = 0; i < N; ++i)
27          cin >> xs[i];
28
29      auto ans = solve(xs);
30
31      if (ans.empty())
32          cout << 0 << '\n';
33      else
34          for (size_t i = 0; i < ans.size(); ++i)
35              cout << ans[i] << (i + 1 == ans.size() ? '\n' : ' ');
36  }
37
38  return 0;
39 }
```