

# AtCoder Beginner Contest 094

## Problema D: *Binomial Coefficients*

---

Prof. Edson Alves - UnB/FGA

Let  $\text{comb}(n, r)$  be the number of ways to choose  $r$  objects from among  $n$  objects, disregarding order. From  $n$  non-negative integers  $a_1, a_2, \dots, a_n$ , select two numbers  $a_i > a_j$  so that  $\text{comb}(a_i, a_j)$  is maximized. If there are multiple pairs that maximize the value, any of them is accepted.

## Constraints

- $2 \leq n \leq 10^5$
- $0 \leq a_i \leq 10^9$
- $a_1, a_2, \dots, a_n$  are pairwise distinct.
- All values in input are integers.

## Input

Input is given from Standard Input in the following format:

```
 $n$   
 $a_1$   $a_2$   $\dots$   $a_n$ 
```

## Output

Print  $a_i$  and  $a_j$  that you selected, with a space in between.

## Exemplos de entradas e saídas

**Entrada**

5  
6 9 4 2 11

**Saída**

11 6

2  
100 0

100 0

## Solução com complexidade $O(N)$

- Este problema pode ser resolvido mediante duas importantes observações
- A primeira delas é que, para um inteiro não-negativo  $m$  fixo,

$$\binom{i}{m} < \binom{j}{m}$$

para  $i < j$ ,  $m \leq i, j$

- Isto significa que, para uma coluna fixa, quanto maior a linha do Triângulo de Pascal, maior o valor do coeficiente binomial correspondente
- A segunda observação é que, para uma linha  $n$  fixa, os coeficientes formam uma sequência crescente até o coeficiente central e segue numa sequência decrescente até o último coeficiente

## Solução com complexidade $O(N \log N)$

- Assim, o coeficiente  $\binom{n}{\lfloor n/2 \rfloor}$  é o maior dentre todos de uma mesma linha e, quanto mais próximo deste centro, maior o coeficiente
- Assim, se os valores da sequência deles forem ordenados, o maior deles será o  $a_i$  procurado
- Para determinar o  $a_j$ , é preciso avaliar os  $N - 1$  termos restantes, em relação à sua distância ao centro: o mais próximo deles é o  $a_j$  desejado
- Esta solução tem complexidade  $O(N \log N)$

## Solução com complexidade $O(N \log N)$

```
5 pair<int, int> solve(vector<int>& as) {  
6     sort(as.begin(), as.end());  
7  
8     auto ai = as.back(), aj = -1, dist = 2000000010;  
9     as.pop_back();  
10  
11     for (auto a : as) {  
12         auto k = min(a, ai - a);  
13  
14         if (ai/2 - k < dist) {  
15             aj = a;  
16             dist = ai/2 - k;  
17         }  
18     }  
19  
20     return { ai, aj };  
21 }
```