

OBI 2012 - Nível 1: Fase 1

Consecutivos

Prof. Edson Alves

Faculdade UnB Gama

Num sorteio que distribui prêmios, um participante inicialmente sorteia um inteiro N e depois N valores. O número de pontos do participante é o tamanho da maior sequência de valores consecutivos iguais. Por exemplo, suponhamos que um participante sorteia $N = 11$ e, nesta ordem, os valores

30, 30, 30, 30, 40, 40, 40, 40, 40, 30, 30

Então, o participante ganha 5 pontos, correspondentes aos 5 valores 40 consecutivos. Note que o participante sorteou 6 valores iguais a 30, mas nem todos são consecutivos.

Sua tarefa é ajudar a organização do evento, escrevendo um programa que determina o número de pontos de um participante.

Entrada

A primeira linha da entrada contém um inteiro N , o número de valores sorteados. A segunda linha contém N valores, V_1, V_2, \dots, V_N , na ordem de sorteio, separados por um espaço em branco.

Saída

Seu programa deve imprimir apenas uma linha, contendo apenas um inteiro, indicando o número de pontos do participante.

Restrições

- ▶ $1 \leq N \leq 10^4$
- ▶ $-2^{31} \leq V_i \leq 2^{31} - 1$, para $i = 1, 2, \dots, N$

Exemplo de entrada e saída

Exemplo de entrada e saída

11

Exemplo de entrada e saída

11



de valores sorteados

Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30

Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30



Números sorteados

Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30

92

83

98

Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30

$$C = 70$$

92

83

98

Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30

$$C = 70$$

92

83

98

Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30

$C = 99$

92

83

98

Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30

$C = 99$

92

83

98

Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30

$$C = 90$$

92

83

98

Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30

$$C = 90$$

92

83

98

Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30

$$C = 98$$

92

83

98

Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30

$$C = 98$$

92

83

98

Exemplo de entrada e saída

11

30 30 30 40 40 40 40 40 30 30 30

$C = 98$



98

92

83

98

Solução: Subtarefa 2 ($K = 1$)

Solução: Subtarefa 2 ($K = 1$)

- ★ Nesta subtarefa, pelo menos um candidato deve ser aprovado

Solução: Subtarefa 2 ($K = 1$)

- ★ Nesta subtarefa, pelo menos um candidato deve ser aprovado
- ★ Seja M a maior nota obtida entre todos os candidatos

Solução: Subtarefa 2 ($K = 1$)

- ★ Nesta subtarefa, pelo menos um candidato deve ser aprovado
- ★ Seja M a maior nota obtida entre todos os candidatos
- ★ Se a nota de corte for igual a M , ao menos o candidato que obteve M será aprovado

Solução: Subtarefa 2 ($K = 1$)

- ★ Nesta subtarefa, pelo menos um candidato deve ser aprovado
- ★ Seja M a maior nota obtida entre todos os candidatos
- ★ Se a nota de corte for igual a M , ao menos o candidato que obteve M será aprovado
- ★ Se a nota de corte for maior que M , ninguém será aprovado

Solução: Subtarefa 2 ($K = 1$)

- ★ Nesta subtarefa, pelo menos um candidato deve ser aprovado
- ★ Seja M a maior nota obtida entre todos os candidatos
- ★ Se a nota de corte for igual a M , ao menos o candidato que obteve M será aprovado
- ★ Se a nota de corte for maior que M , ninguém será aprovado
- ★ Portanto, para esta subtarefa a resposta é a maior nota obtida entre todos os candidatos

Solução: Subtarefa 4 ($A_i \leq 2$)

Solução: Subtarefa 4 ($A_i \leq 2$)

★ Nesta subtarefa, todos candidatos tiraram ou nota 1 ou nota 2

Solução: Subtarefa 4 ($A_i \leq 2$)

- ★ Nesta subtarefa, todos candidatos tiraram ou nota 1 ou nota 2
- ★ Só há duas alternativas para a nota de corte: $C = 1$ e $C = 2$

Solução: Subtarefa 4 ($A_i \leq 2$)

- ★ Nesta subtarefa, todos candidatos tiraram ou nota 1 ou nota 2
- ★ Só há duas alternativas para a nota de corte: $C = 1$ e $C = 2$
- ★ Se a nota de corte for igual a 1, todos serão aprovados

Solução: Subtarefa 4 ($A_i \leq 2$)

- ★ Nesta subtarefa, todos candidatos tiraram ou nota 1 ou nota 2
- ★ Só há duas alternativas para a nota de corte: $C = 1$ e $C = 2$
- ★ Se a nota de corte for igual a 1, todos serão aprovados
- ★ Se a nota de corte for igual a 2, apenas os candidatos que tiraram 2 serão aprovados

Solução: Subtarefa 4 ($A_i \leq 2$)

- ★ Nesta subtarefa, todos candidatos tiraram ou nota 1 ou nota 2
- ★ Só há duas alternativas para a nota de corte: $C = 1$ e $C = 2$
- ★ Se a nota de corte for igual a 1, todos serão aprovados
- ★ Se a nota de corte for igual a 2, apenas os candidatos que tiraram 2 serão aprovados
- ★ Portanto, a resposta só será 2 quanto o número de candidatos que tiraram 2 for maior ou igual a K ; caso contrário, a resposta é igual a 1

Solução: Subtarefa 3 ($K = 3$)

Solução: Subtarefa 3 ($K = 3$)

- ★ Nesta subtarefa deve ser classificados, no mínimo, 3 candidatos

Solução: Subtarefa 3 ($K = 3$)

- ★ Nesta subtarefa deve ser classificados, no mínimo, 3 candidatos
- ★ Na Subtarefa 1 a resposta era a nota do primeiro classificado

Solução: Subtarefa 3 ($K = 3$)

- ★ Nesta subtarefa deve ser classificados, no mínimo, 3 candidatos
- ★ Na Subtarefa 1 a resposta era a nota do primeiro classificado
- ★ Nesta subtarefa, a resposta será a nota do terceiro colocado no concurso

Solução: Subtarefa 3 ($K = 3$)

- ★ Nesta subtarefa deve ser classificados, no mínimo, 3 candidatos
- ★ Na Subtarefa 1 a resposta era a nota do primeiro classificado
- ★ Nesta subtarefa, a resposta será a nota do terceiro colocado no concurso
- ★ Para determinar esta nota, é preciso manter o registro das três maiores notas observadas até o momento

Solução: Subtarefa 3 ($K = 3$)

- ★ Nesta subtarefa deve ser classificados, no mínimo, 3 candidatos
- ★ Na Subtarefa 1 a resposta era a nota do primeiro classificado
- ★ Nesta subtarefa, a resposta será a nota do terceiro colocado no concurso
- ★ Para determinar esta nota, é preciso manter o registro das três maiores notas observadas até o momento
- ★ A cada iteração do laço, as variáveis x , y e z serão atualizadas para que representem a terceira, a segunda e a primeira melhor nota, respectivamente

Solução

Solução

- ★ Para resolver o caso geral do problema, é preciso armazenar todas as notas em um vetor

Solução

- ★ Para resolver o caso geral do problema, é preciso armazenar todas as notas em um vetor
- ★ A nota do i -ésimo candidato será dada por $As[i]$, sendo que a contagem começa em zero

Solução

- ★ Para resolver o caso geral do problema, é preciso armazenar todas as notas em um vetor
- ★ A nota do i -ésimo candidato será dada por $As[i]$, sendo que a contagem começa em zero
- ★ A estratégia de solução será uma busca completa em C

Solução

- ★ Para resolver o caso geral do problema, é preciso armazenar todas as notas em um vetor
- ★ A nota do i -ésimo candidato será dada por $As[i]$, sendo que a contagem começa em zero
- ★ A estratégia de solução será uma busca completa em C
- ★ Iniciando em $C = 100$, serão avaliadas as possíveis notas de corte, em ordem decrescente, até encontrar o primeiro valor para o qual o número de aprovados é maior ou igual a K

Solução

- ★ Para resolver o caso geral do problema, é preciso armazenar todas as notas em um vetor
- ★ A nota do i -ésimo candidato será dada por $As[i]$, sendo que a contagem começa em zero
- ★ A estratégia de solução será uma busca completa em C
- ★ Iniciando em $C = 100$, serão avaliadas as possíveis notas de corte, em ordem decrescente, até encontrar o primeiro valor para o qual o número de aprovados é maior ou igual a K
- ★ Este solução tem complexidade $O(NC)$

```
#include <stdio.h>
```

```
int main()  
{
```

```
    if (consecutivos > resposta)  
        resposta = consecutivos;
```

```
#include <stdio.h>
```

```
int main()  
{  
    int N;  
    scanf("%d", &N);
```

```
if (consecutivos > resposta)  
    resposta = consecutivos;
```

```
#include <stdio.h>
```

```
int main()  
{  
    int N;  
    scanf("%d", &N);  
  
    int V[N];
```

```
    if (consecutivos > resposta)  
        resposta = consecutivos;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

```
    for (int i = 0; i < N; ++i)
```

```
        scanf("%d", &V[i]);
```

```
    if (consecutivos > resposta)
```

```
        resposta = consecutivos;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

```
    for (int i = 0; i < N; ++i)
```

```
        scanf("%d", &V[i]);
```

```
    int v = V[0], consecutivos = 1, resposta = 1;
```

```
        if (consecutivos > resposta)
            resposta = consecutivos;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

```
    for (int i = 0; i < N; ++i)
```

```
        scanf("%d", &V[i]);
```

```
    int v = V[0], consecutivos = 1, resposta = 1;
```

```
    for (int i = 1; i < N; ++i)
```

```
        if (consecutivos > resposta)
            resposta = consecutivos;
```



```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

```
    for (int i = 0; i < N; ++i)
```

```
        scanf("%d", &V[i]);
```

```
    int v = V[0], consecutivos = 1, resposta = 1;
```

```
    for (int i = 1; i < N; ++i)
```

```
        if (V[i] == v)
```

```
            ++consecutivos;
```

```
        if (consecutivos > resposta)
```

```
            if (consecutivos > resposta)
                resposta = consecutivos;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

```
    for (int i = 0; i < N; ++i)
```

```
        scanf("%d", &V[i]);
```

```
    int v = V[0], consecutivos = 1, resposta = 1;
```

```
    for (int i = 1; i < N; ++i)
```

```
        if (V[i] == v)
```

```
            ++consecutivos;
```

```
        else
```

```
        {
```

```
            if (consecutivos > resposta)
```

```
                if (consecutivos > resposta)
                    resposta = consecutivos;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

```
    for (int i = 0; i < N; ++i)
```

```
        scanf("%d", &V[i]);
```

```
    int v = V[0], consecutivos = 1, resposta = 1;
```

```
    for (int i = 1; i < N; ++i)
```

```
        if (V[i] == v)
```

```
            ++consecutivos;
```

```
        else
```

```
        {
```

```
            if (consecutivos > resposta)
```

```
                v = V[i];
```

```
        }
```

```
    if (consecutivos > resposta)
```

```
        resposta = consecutivos;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

```
    for (int i = 0; i < N; ++i)
```

```
        scanf("%d", &V[i]);
```

```
    int v = V[0], consecutivos = 1, resposta = 1;
```

```
    for (int i = 1; i < N; ++i)
```

```
        if (V[i] == v)
```

```
            ++consecutivos;
```

```
        else
```

```
        {
```

```
            if (consecutivos > resposta)
```

```
                v = V[i];
```

```
                consecutivos = 1;
```

```
        }
```

```
    if (consecutivos > resposta)
```

```
        resposta = consecutivos;
```

```
#include <stdio.h>
```

```
int main()
```

```
{
```

```
    int N;
```

```
    scanf("%d", &N);
```

```
    int V[N];
```

```
    for (int i = 0; i < N; ++i)
```

```
        scanf("%d", &V[i]);
```

```
    int v = V[0], consecutivos = 1, resposta = 1;
```

```
    for (int i = 1; i < N; ++i)
```

```
        if (V[i] == v)
```

```
            ++consecutivos;
```

```
        else
```

```
        {
```

```
            if (consecutivos > resposta)
```

```
                v = V[i];
```

```
                consecutivos = 1;
```

```
        }
```

```
    }
```

```
    if (consecutivos > resposta)
```

```
        resposta = consecutivos;
```

Bônus

Bônus

- ★ Há uma solução mais eficiente para este problema

Bônus

- ★ Há uma solução mais eficiente para este problema
- ★ Por meio da ordenação das notas, é possível adotar uma estratégia gulosa

Bônus

- ★ Há uma solução mais eficiente para este problema
- ★ Por meio da ordenação das notas, é possível adotar uma estratégia gulosa
- ★ A resposta será a nota do K -ésimo aprovado

Bônus

- ★ Há uma solução mais eficiente para este problema
- ★ Por meio da ordenação das notas, é possível adotar uma estratégia gulosa
- ★ A resposta será a nota do K -ésimo aprovado
- ★ Se o vetor As for ordenado em ordem crescente, o K -ésimo aprovado ocupará o índice $N - K$ do vetor

Bônus

- ★ Há uma solução mais eficiente para este problema
- ★ Por meio da ordenação das notas, é possível adotar uma estratégia gulosa
- ★ A resposta será a nota do K -ésimo aprovado
- ★ Se o vetor As for ordenado em ordem crescente, o K -ésimo aprovado ocupará o índice $N - K$ do vetor
- ★ Este solução tem complexidade $O(N \log N)$