

MC458A - Laboratório 1: Ana Sabi Tudor e suas paradas

1 Introdução

A lendária ex-aluna de computação da UNICOMP, Ana Sabi Tudor, gostava muito de viajar pelos países ibéricos (Portugal e Espanha) durante as férias. Ana tem uma bicicleta dobrável que ela carrega quando pega um ônibus durante a viagem. Em toda viagem ela seguia o mesmo percurso que contém vários pontos de parada. Em um ponto de parada qualquer ela pode (i) seguir para o próximo ponto de parada de bicicleta, se ela considerar o trecho agradável ou (ii) pegar um ônibus que vai para a próximo ponto de parada.

Após várias viagens ao longo dos anos, Ana conseguiu avaliar cada trecho entre dois pontos de parada em uma escala (de valores inteiros) de *niceness* (“*agradabilidade*”). Nesta escala, um valor positivo indica o quanto Ana gosta daquele trecho e um valor negativo o quanto ela detesta; um valor 0 (zero) indica que Ana é indiferente àquele trecho. O valor de cada trecho é baseado tanto na experiência pessoal da Ana como em outros fatores tais como opiniões de amigas ou *reviews* de sites.

Ana decidiu que na próxima viagem (que ela espera que esteja bem próxima) ela usará a bicicleta apenas em uma sequência consecutiva de trechos. Ela não se importa de andar em trechos que ela não gosta contanto que eles liguem trechos que ela goste o suficiente para compensar. É possível que na nova escala, nenhum trecho seja agradável o suficiente e neste caso ela faria a viagem inteira de ônibus.

Como Ana está meio enferrujada em programação, ela gostaria que você fizesse um programa para ajudá-la a planejar sua viagem.

2 Especificação de entrada e saída

A entrada é composta por duas linhas:

- A primeira linha contém um inteiro n ($10 \leq n \leq 50000$) que representa o número de pontos de parada.
- A segunda linha contém $n - 1$ inteiros p_1, \dots, p_{n-1} ($-1000 \leq p_i \leq 1000$) em que p_i é o *niceness* do trecho entre o ponto i e o ponto $i + 1$.

Você deve determinar um ponto inicial i e um ponto final j que identificam a sequência de trechos cuja soma $p_i + p_{i+1} + \dots + p_j$ é a maior possível. Se houver mais que uma solução, você deve escolher uma que maximiza $j - i$. Se ainda assim houver mais que uma solução você deve escolher aquela com menor i .

A saída do seu programa deve ser uma linha contendo os valores i e j . Se não houver uma sequência com soma positiva, deve-se imprimir 0 e 0. Note que os pontos de paradas são indexados a partir do 1.

Observação: não será exigido, mas sua implementação deveria ter complexidade $O(n)$. Alguns testes não permitirão uma implementação de complexidade $\Omega(n^2)$ (limite de tempo).

Exemplo:

Entrada	Saída
10 -4 9 8 5 -2 7 -1 -2 -8	2 6

Entrada	Saída
10 -4 -9 -8 -5 -2 -7 -1 -2 -8	0 0

3 Implementação e Submissão

- A solução deverá ser implementada em C, C++, ou Python 3. Só é permitido o uso de bibliotecas padrão. **Não é permitido o uso de bibliotecas que não sejam padrão e de flags/diretivas de otimização.**
- O programa deve ser submetido no SuSy, com o nome principal **t1** (por exemplo, t1.c).
- O número máximo de submissões é 20.
- A tarefa contém 10 testes abertos e 10 testes fechados. A nota será proporcional ao número de acertos nos testes fechados.

A solução pode ser submetida até o dia 01/04/24 às 08:00.