

MC458A - Laboratório 4: processamento de dados

1 Introdução

Parabéns! Você conseguiu o emprego dos seus sonhos (ou não...). Você é o responsável por um sistema de computação de alto desempenho que é capaz de processar terabytes de dados por dia. Por um período de n dias você recebe uma quantidade de dados: no dia i você recebe x_i terabytes. Para cada terabyte que você processa, você ganha um lucro fixo, mas qualquer dado não-processado torna-se indisponível ao fim do dia (ou seja, você não pode mais processá-lo no futuro).

Você não consegue processar tudo cada dia por causa das limitações do seu sistema computacional (You are not Google!), o qual consegue processar uma quantidade fixa de terabytes em um dado dia. De fato, o sistema roda um software um tanto exótico que, embora seja sofisticado, não é totalmente confiável (Maybe you are Google...). Assim, a quantidade de dados que você consegue processar vai diminuindo a cada dia que passa desde a última vez que um reboot foi feito. No primeiro dia de um reboot, o sistema consegue processar s_1 terabytes, no segundo dia após o reboot, o sistema consegue processar s_2 terabytes e assim por diante até s_n ; suponha que $s_1 > s_2 > s_3 > \dots > s_n > 0$. (Naturalmente, no dia i o sistema só pode processar até x_i terabytes, independente de quão rápido o sistema seja.) Para recolocar o sistema no máximo desempenho, você pode optar por fazer um reboot em qualquer dia. Entretanto em qualquer dia que você decida fazer um reboot, o sistema não pode processar nenhum dado naquele dia.

Problema. Dadas as quantidades de dados disponíveis x_1, x_2, \dots, x_n para os próximos n dias e dada a configuração do seu sistema expressa por s_1, s_2, \dots, s_n (e supondo que foi feito um reboot no dia 0), decida os dias nos quais você fará os reboots para maximizar a quantidade total de dados processados.

Exemplo. Suponha que $n = 4$ e os valores de x_i e s_i são dados pela tabela seguinte.

Dia	1	2	3	4
x	10	1	7	7
s	8	4	2	1

A melhor solução seria fazer um reboot apenas no dia 2. Deste modo, o sistema processa 8 terabytes no dia 1, 0 no dia 2, 7 no dia 3 e 4 no dia 4, totalizando 19. Note que se você não fizer nenhum reboot, o sistema processaria $8 + 1 + 2 + 1 = 12$; qualquer outra estratégia de reboot resultaria em menos que 19 terabytes.

2 Especificação de entrada e saída

A entrada é composta por 3 linhas:

- A primeira linha contém um inteiro n ($1 \leq n \leq 1000$) que representa o número de dias.
- A segunda linha contém os valores x_1, x_2, \dots, x_n de dados a serem recebidos em cada dia.
- A terceira linha contém a configuração do sistema, ou seja, os valores s_1, s_2, \dots, s_n de dados que você é capaz de processar a cada dia depois de um reboot.

A saída deve conter 2 linhas:

- A primeira linha deve conter o número máximo de dados capaz de processar no período de tempo.
- A segunda linha deve conter os dias em que é necessário realizar o reboot de modo a maximizar a quantidade de dados processada.

Considere que existe apenas uma única solução ótima, ou seja, existe uma única escolha de dias para reboot que maximiza a quantidade de dados processadas. Caso a solução ótima envolva não realizar nenhum reboot, indique que é realizado um reboot apenas no dia 0.

Exemplo:

Entrada	Saída
4	19
10 1 7 7	0 2
8 4 2 1	

Entrada	Saída
6	60
20 18 2 22 1 19	0 3 5
16 12 10 8 4 2	

3 Implementação e Submissão

- A solução deverá ser implementada em C, C++, ou Python 3. 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 **t4** (por exemplo, t4.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 28/05/24, às 8hrs.