

PROBLEMA

Duas pessoas, Marcelo e Carla, trabalham em uma fazenda de maçãs na qual existem N árvores. As árvores estão alinhadas em uma fileira e são numeradas de 1 a N . Marcelo planeja coletar maçãs de K árvores consecutivas e Carla de L árvores consecutivas. Eles querem escolher segmentos disjuntos, ou seja, que não se sobrepõem para não interferirem na coleta do outro. Qual o maior número de maçãs que os dois juntos podem coletar?

Escreva uma função:

def get_max_apples(A, K, L)

que, dado um vetor A consistindo de N inteiros denotando o número de maçãs em cada árvore, e inteiros K e L , respectivamente, o número de árvores que Marcelo e Carla escolhem coletar, retorne o número máximo de maçãs coletadas ou -1 se não existem intervalos que permitam a coleta.

Por exemplo, dado um vetor $A = [3, 4, 1, 7, 8, 5]$, $K = 2$, $L = 3$, sua função deve retornar 27, já que Carla escolherá as árvores 4, 5 e 6 e coletar $7 + 8 + 5 = 20$ maçãs. Marcelo, por sua vez, escolherá as árvores 1 e 2, coletando assim $3 + 4 = 7$ maçãs.

SOLUÇÃO

* BACKEND

Partindo de um olhar técnico, podemos observar que os números inteiros K e L , representam o número de elementos de subconjuntos de A , onde o somatório dos elementos dos 2 conjuntos deve ser o maior possível.

Sendo assim podemos a partir de uma matriz, representar a soma de todos os elementos dos subconjuntos possíveis, de tamanhos K e L , onde os subconjuntos de tamanho K representarão as linhas, e os subconjuntos de tamanho L representarão as colunas.

Exemplo:

Para,

$A = [3, 4, 1, 7, 8, 5]$

$K = 2$

L = 3,

temos:

	[3,4,1]	[4,1,7]	[1,7,8]	[7,8,5]
[3,4]	x	x	23	27
[4,1]	x	x	x	25
[1,7]	x	x	x	x
[7,8]	23	x	x	x
[8,5]	21	25	x	x

A matriz acima representa o somatório entre os subconjuntos, as células que possuem um **x** significa que não é possível obter uma soma, pois os subconjuntos se sobrepõem.

A partir dessa analogia, podemos obter o seguinte algoritmo:

maior = -1

para i de 1 até K faça

para j de 1 até L faça

se (j >= i + K) ou (i > (j+L)) então

acumulador = 0

para x de i até i+K faça

acumulador += A[x]

fimpara

para x de j até j+L faça

acumulador += A[x]

fimpara

se maior < acumulador então

maior = acumulador

fimse

fimse

fimpara

fimpara

Não foi necessário criar de fato uma matriz, pois a partir dos somatórios, conseguimos obter o de maior valor, em tempo de execução e indica-lo.

No trecho "*se $(j \geq i + K)$ ou $(i > (j+L))$ então*" verificamos se é possível fazer o somatório daquela célula, pois os subconjuntos não podem se sobrepor. Isso foi possível pois os índices i e j representam o início daquele subconjunto dentro de A , ex:

O subconjunto $[7,8]$, está na linha 4 da matriz, e dentro do conjunto A , o número 7 está na posição 4.

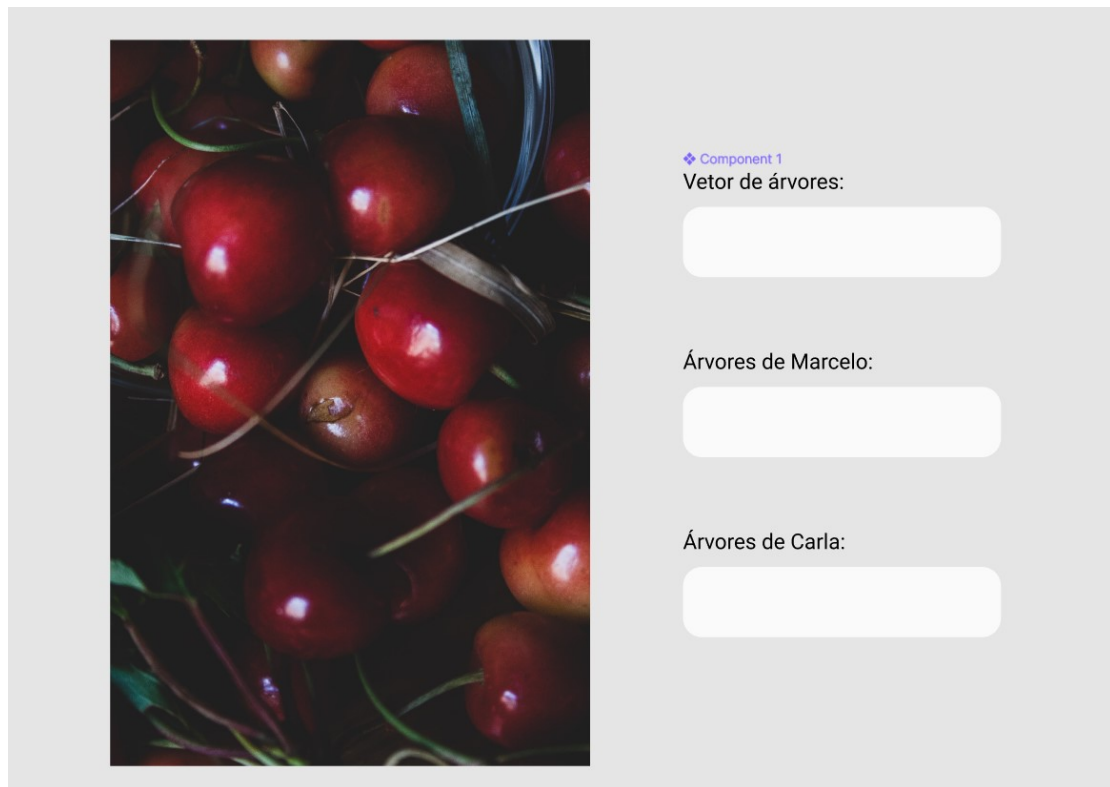
Já o subconjunto da coluna 1 da matriz, representado por $[3,4,1]$, dentro do conjunto A , o número 3 está na posição 1.

A partir disso foi necessário apenas fazer a relação com o número de elementos de K e de L , para saber se o número de elementos passavam do início i ou j .

***FRONTEND**

Utilizando o figma

Tela inicial



Tela do resultado

Resultado

Número de maçãs por pé



Número de maçãs colhidas por Marcelo



Número de maçãs colhidas por Carla



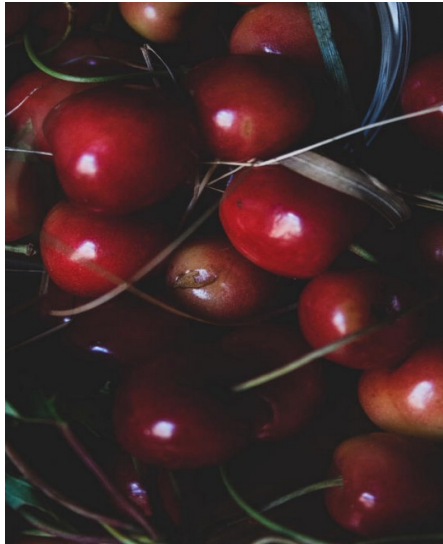
Total: XX

Na tela inicial temos as seguintes entradas:

1. Vetor de árvores = deve ser no formato: $X_1, X_2, X_3, \dots, X_n$
2. Árvores de marcelo = Deve ser um inteiro > 0
3. Árvores de Carla = Deve ser um inteiro > 0

RESULTADO DAS TELAS EM PRÁTICA

TELA INICIAL 2



Consulte a quantidade máxima de maçãs colhidas!

Vetor de árvores:

4,5,2,3,4,5

Árvores de Marcelo:

3

Árvores de Carla:

2

Calcular

TELA RESULTADO 2

