Soma Máxima

Motivação

Dado um vetor unidimensional o algoritmo desenvolvido é capaz de encontrar a maior soma possível em um sub-vetor contíguo, assim como os índices referentes à tal soma. Esse tipo de problema é encontrado em diversas aplicações reais como:

Processamento de imagens: pode ser utilizado para detectar as regiões com os maiores valores de um vetor de uma imagem, às quais representam as áreas de maior brilho.

Análise sequencial de genoma: pode ser aplicado para encontrar importantes sequências biológicas de proteína, facilitando o entendimento de sua estrutura.

Dentre outras.

Exemplo de Execução

Input:
10
31 -41 59 26 -53 58 97 -93 -23 84

Output:
Soma: 187
Índices: 3 a 7

O algoritmo

Para facilitar a compreensão, aqui estão algumas simplificações:

- *SMt* = soma máxima total, corresponde a soma máxima de um sub-vetor
- *SMi* = soma máxima possível que termine no índice i.
- A = vetor com os valores de entrada

Inicialmente armazenei os n valores inteiros inseridos pelo usuário em um array de tamanho 20, portanto utilizarei apenas os n primeiros valores, haja vista a especificação do problema tratar de arrays de tamanho 3 a 20.

Maior Soma

Para encontrar a maior soma de um sub-vetor, primeiro devemos pensar no menor valor que a *SMt* pode assumir, que seria em um vetor composto apenas por números negativos ou então um vetor vazio, nesse caso a *SMt* de um sub-vetor seria 0. Portanto inicializei a variável de *SMi* e *SMt* como zero.

Após o passo acima foi realizado uma varredura da esquerda para direita do vetor visando encontrar todas as *SMi's*. Dessa forma, a *SMi* para algum i seria A[i] + Sm(i-1) caso SM(i-1) seja ≥ 0. Portanto de forma recursiva e utilizando como caso base *SMi* = 0 podemos obter todos *SMi's*.

$$SM0 = 0$$

$$SMi = A[i] + SM(i-1), \text{ para } SM(i-1) \geq 0$$

$$SMi = A[i], \text{ para } SM(i-1) < 0$$

Como estamos interessados apenas na *SMt*, compara-se em cada laço de repetição se a *SMi* é maior que a *SMt* e caso seja uma afirmação verdadeira, então:

$$SMt = SMi.$$

Índices da Maior Soma

Para encontrar os índices referentes ao intervalo de *SMt*, sempre que a *SMi* > *SMt* então devemos igualar o índice de fim da *SMt* a *i.* Quanto ao índice de início sempre que a *SM(i-1)* < 0 então devemos mudar o índice de início de *SMi*, no entanto apenas quando *SMi* for maior que a *SMt* igualamos o índice de início da *SMt* ao índice de início de *SMi*.

Análise Assintótica

No pior caso: O(n), o custo de tempo é sempre proporcional ao tamanho do vetor, pois são feitas duas comparações em todos os n's valores. O(n), o custo de espaço é sempre o tamanho do array mais as 5 variáveis utilizadas no algoritmo.

Quadrado Mágico

O algoritmo

Para simplificar o problema dividi os quadrados mágicos em ímpares, ou seja, possuem dimensão n x n sendo que n é um número ímpar, e em duplamente ou singularmente par, ou seja, n é um número múltiplo de 2 e 4 ou apenas múltiplo de 4 respectivamente.

Quadrados mágicos ímpares

Para gerar um quadrado mágico dessa ordem é necessário primeiro posicionar o número 1 para definir qual das quatro possíveis rotações do quadrado mágico será gerada. Ele pode ser posicionado em uma posição que haja poucas somas pois é um dos números que possuem a menor quantidade de somas que resultam na constante mágica.

Os demais valores são alocados em ordem crescente, utilizando as seguintes regras:

- O posicionamento natural é feito um quadrado à direita e à cima.
- Caso saia dos limites do quadrado pelas laterais, então retorna-se à outra lateral.
- Caso saia dos limites do quadrado mágico na vertical, então retorna-se para o lado oposto de saída.

- Caso o próximo posicionamento esteja ocupado, então retorne a posição anterior e aloque o número no quadrado abaixo.

Seguindo essas regras podemos obter quadrados mágicos 3x3 e 5x5 em que cada coluna, linha e diagonal resultam na constante mágica 15 e 65 respectivamente.

Análise Assintótica

No pior caso: $O(n^2)$ é o custo de tempo, pois para um quadrado mágico de tamanho n é necessário n*n cálculos para determinar o valor atribuído à cada posição, já no custo de espaço é O(1), pois é necessário apenas armazenar a dimensão do quadrado mágico para que se possa gerá-lo seguindo a lógica de montagem.

Quadrados mágicos singularmente pares

Para resolver esse problema estamos preocupados em somente no caso de quadrados mágicos de tamanho 6x6, inicialmente dividimos esse quadrado maior em outros quatro quadrados menores de tamanho 3x3 e resolvemos os quadrados mágicos como se fossem ímpares, cada um responsável por um quarto dos números.

Após feito essa etapa, realizamos 3 trocas de posições para que finalmente o quadrado se torne mágico, obtendo a soma das linhas, colunas e diagonais equivalente a 111.

Quadrados mágicos duplamente pares

Para este caso estamos interessados apenas no quadrado mágico de tamanho 4x4, inicialmente enumeramos da esquerda pra direita todo o quadrado em ordem crescente, após essa etapa trocamos os valores dos cantos e o quadrado 2x2 do centro para como se estivéssemos enumerando o quadrado mágico em ordem decrescente.

Obtendo assim um quadrado 4x4 com colunas, linhas e diagonais com a soma igual a 34.