

Trabalho Prático 2 - Algoritmos I

Avaliação do evento

Davi Brito

Departamento de Ciência da Computação - Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte - MG - Brasil
daviembrito@ufmg.br

1. Introdução

O problema proposto envolve calcular a sequência de shows do Rock in Rio que vale a pena assistir futuramente, se baseando nas notas de amigos nos shows de um ano anterior. Para isso, cada amigo dará uma nota de -5 a 5 para cada show do festival. A partir daí, as notas finais de cada show serão utilizadas para cálculo da melhor subsequência.

O programa criado permite ler cada nota dada e calcular as somas totais para cada show. Posteriormente, esses dados são utilizados para se calcular a subsequência contínua de soma máxima das notas dos shows. O número inicial e final dos shows da subsequência escolhida será a saída gerada.

2. Modelagem

2.1. Estruturas de dados

O programa foi implementado com a utilização da estrutura de dados *vector*, implementada na biblioteca padrão da linguagem, devido a sua facilidade de inicialização e tempo constante de acesso a posições.

Por meio da estrutura usada, foram armazenadas as somas das notas dadas a cada show pelos amigos. A partir dela, o algoritmo principal realizou cálculos e retornou seu resultado novamente em um *vector* de números flutuantes.

2.2. Algoritmos e complexidades

Para resolver o problema proposto, o programa utilizou-se de um algoritmo de solução de soma máxima de sublista contígua por meio de divisão e conquista.

Inicialmente, as notas de cada pessoa foram somadas a um vetor de S posições inicializado com valores nulos, sendo S o número de shows para o caso dado. Ao final da leitura das entradas era retornado o vetor final, que continha o somatório de cada nota individual dos A amigos, agora agregadas de acordo com cada show correspondente. Esse processo de leitura possui complexidade temporal pertencente a $O(A \cdot S)$.

Em seguida, o vetor de somas totais é utilizado como parâmetro para o algoritmo principal.

O algoritmo consiste em dividir o vetor original em subvetores com metade do tamanho cada um, calcular recursivamente o subvetor de soma máxima do lado esquerdo, direito e da junção deles, e, por fim, retornar o maior valor desses três.

O caso base da recursão se dá quando o índice de início passado é igual ao índice de fim do subvetor, indicando que há somente um elemento no conjunto. Assim, os valores retornados serão os próprios índices e o valor guardado naquela posição

A função que calcula a soma máxima cruzada dos subvetores atua olhando até qual índice da parte esquerda a soma dá um valor maior, realizando a mesma operação também para o lado direito. Por fim, são retornados os índices do maior subvetor cruzado, assim como o valor dessa soma.

O algoritmo possui complexidade de tempo pertencente a $O(n \log n)$, sendo n o tamanho do vetor passado. Esse valor é dado pelas duas chamadas recursivas realizadas, além da chamada de função para cálculo da soma cruzada. É denotado pela seguinte relação de recorrência:

$$T(n) = 2T(n/2) + O(n)$$

3. Conclusão

O trabalho consistiu em implementar um programa de cálculo de subvetor contíguo de soma máxima, no contexto de notas dadas a shows do festival de música *Rock in Rio*, utilizando o paradigma de divisão e conquista.

A partir desse trabalho prático, foi possível pôr em prática conhecimentos ligados ao paradigma de programação utilizado, além de revisar conceitos de recursividade e boas práticas de programação.

4. Instruções de compilação e execução

- Acessar a pasta `/tp2/`
- Utilizar o comando `make` no terminal.
- Rodar o programa com `./tp2 < entrada.txt`
- A saída será dada na interface de comandos