

Trabalho Prático 2

Avaliação do evento

Vitor Emanuel Ferreira Vital - 2021032072

Departamento de Ciência da Computação - Universidade Federal de Minas Gerais (UFMG)
Belo Horizonte - MG - Brazil

vitorvital@dcc.ufmg.br

1. Introdução

Esta documentação tem como objetivo apresentar a implementação e modelagem da resolução do Trabalho Prático 2 da disciplina de Algoritmos 1. Em tal prática, os principais temas explorados, solucionados na linguagem C++, são:

- Resolução de problemas reais via modelos algorítmicos
- Utilização de estruturas de dados estudados na disciplina
- Utilização do método de divisão e conquista

O problema a ser tratado lida com a programação dos shows que um grupo de amigos estará presente, no qual, dado um conjunto de avaliações dos shows, com base em experiências passadas, o grupo gostaria de saber qual o melhor conjunto consecutivo de shows é melhor estar. Com o objetivo de uma melhor experiência, visto que ir em shows com péssimas avaliações seria ruim para os amigos, encontraremos o melhor conjunto de shows para eles.

A documentação explicita os métodos e modelagens utilizados para resolução do problema em questão.

2. Modelagem

Conforme explicitado, cada um dos amigos devem avaliar os shows com base em experiências passadas, sendo essa nota pertencente ao conjunto dos reais $[-5,5]$. Dentre todas as escolhas realizadas, o programa deve somar as avaliações para cada show e encontrar o subvetor com maior soma possível, mantendo as seguintes características:

1. Eles sempre optam pela maior quantidade de shows possível, ou seja, se há uma avaliação cuja soma resulta em 0, eles a incluirão se puderem.
2. Ao sair do evento eles não poderão retornar.

Com isso em mãos podemos fazer a modelagem seguindo o problema de encontrar o subvetor de maior soma dado um vetor inicial. Nesse caso, dado um array, deseja-se verificar a maior soma possível.

Para o caso em questão utiliza-se o algoritmo de Divisão e Conquista, esse algoritmo possui comportamento semelhante ao do MergeSort, porém adaptado para o caso se encontrarmos o subvetor de maior soma.

Algoritmo de Divisão e Conquista:

Ao invés de calcular a soma de cada subvetor possível, o que geraria um custo exagerado de tempo, o algoritmo em questão utiliza o cálculo da soma de subgrupos, comparando-os e obtendo ao

final o resultado desejado. A partir do último elemento do vetor calcularmos a soma de cada subvetor possível até obtermos um subvetor unitário.

As divisões são realizadas da seguinte forma:

1. Dividimos o vetor em duas metades, do início ao elemento central e, uma segunda, do elemento central para o mais a direita
2. Além desse, um subvetor de forma que ele cruze o elemento central

Os elementos esquerdo e direito são recursivamente chamados até atingir um vetor unitário, que também pode ser um subvetor com maior soma, caso em uma dessas chamadas encontre-se o subvetor ele será indicado ao final do programa.

Caso a soma máxima do subvetor esteja em um que cruze o elemento central, basta que encontre-se a soma máxima começando no ponto médio e terminando em algum ponto à esquerda dele, posteriormente, encontramos a soma máxima começando do elemento central +1 e terminando em algum ponto à direita dele. Por fim, basta combinar essas soluções, retornando:

1. Subvetor à esquerda - caso seja a maior soma obtida;
2. Subvetor à direita - caso seja a maior soma obtida;
3. A união dos anteriores com o elemento central - caso seja a maior soma obtida.

Com base nessa implementação, precisa-se manter o registro da maior soma obtida, ou seja, ao calcular $\text{SumMax}(A[i] \text{ até } A[j])$, caso possua a maior soma obtida armazena-se ele em uma variável auxiliar com soma global máxima e a posição de início e término desse subvetor dentro do vetor principal. Assim, ao final da execução do programa basta retornar essa posição, pois contém a sequência de shows que maximiza a diversão dos amigos.

Ao fim do problema conclui-se que é possível modelar esse problema como encontrar a o subvetor de soma máxima, pois precisa-se definir a melhor combinação de shows consecutivos no qual os integrantes participarão, assim obtém-se uma modelagem de resolução fácil no qual é plausível utilizar o Algoritmo de Divisão e Conquista, que realiza o cálculo dividindo o vetor inicial em subproblemas relacionados menores para obtenção de uma solução ótima posteriormente. O algoritmo foi utilizado pois, além de utilizar da divisão e conquista, possui custo $O(n \log n)$ de tempo com a entrada.