

TRABALHO PRÁTICO 2

Avaliação do evento

Tarcizio Augusto Santos Lafaiete

Universidade Federal de Minas Gerais(UFMG)
Belo Horizonte - MG - Brasil

tarcizio-augusto@hotmail.com

1.Introdução

O problema proposto para implementação do trabalho prático foi modelar um algoritmo que ajude um grupo de amigos a definir o intervalo de shows consecutivos do Rock In Rio que mais os agrada para assim aproveitarem o festival da melhor forma, uma vez que caso eles saiam do festival não poderão entrar novamente. O programa recebe como entrada as notas de +5 a -5 dada pelos m amigos a cada um dos n shows do festival.

2.Interpretando o problema

Com a base do problema estabelecida, pode-se partir para a interpretação deste problema em requisitos computacionais e algorítmicos. Inicialmente neste problema é dado para nós m vetores contendo as notas dos n shows dadas pelo grupo de amigos, com isso em mente é interessante para o processamento de dados da entrada transformar estes m vetores em um único vetor, para isso realizamos o somatório das m notas de cada show e salvamos na posição correspondente ao show neste vetor unificado.

A partir do vetor que foi gerado deve-se pensar em uma maneira de avaliar as notas dadas pelos m amigos para definir o intervalo que solucione o problema, assim sendo este intervalo será dado pelo intervalo do vetor cujo a soma será máxima dentro do vetor, ou seja a resposta do problema será dada por um subvetor de tamanho c, no qual c será menor ou igual ao tamanho n do vetor unificado, e o somatório de seus elementos será o máximo possível dentre os valores do vetor.

Definido o problema através das afirmações anteriores, pode-se perceber que este problema pode ser interpretado com um problema padrão da computação, o chamado Subvetor de Soma Máxima (Maximum Subarray Sum, em inglês).

3. Modelagem do problema

Como dito no tópico anterior o problema proposto para este trabalho prático se encaixa na descrição de um Subvetor de Soma Máxima, que possui uma solução conhecida. Inicialmente pode-se pensar em solucionar este problema de diversas formas e utilizando-se de vários paradigmas da computação como por exemplo

indução e divisão e conquista. Contudo como especificado no trabalho a versão a ser explorada aqui será utilizando o paradigma da divisão e conquista.

A divisão e conquista é um paradigma da computação que foi formalizado pela primeira vez por Anatolii Karatsuba em 1960, na qual ele descreve esta técnica como uma forma de solucionar problemas os dividindo em subproblemas e “conquistando” a solução de cada subproblema em separado para assim no final combinar estas soluções para se gerar a solução definitiva do problema. Desde de então este paradigma vem sendo utilizado em grande medida como uma poderosa ferramenta para a solução de problemas na computação, como por exemplo a ordenação de vetores(Quick Sort e Merge Sort) e facilitando a computação da transformada discreta de Fourier através da transformada rápida de Fourier.

Esta técnica possui três fases para a solução do problema, a primeira é a divisão na qual pega-se o problema em seu tamanho original e o divide em subproblemas de tamanho menor e assim é feito recursivamente até se chegar em problemas pequenos o suficiente para serem solucionados de forma simples. O segundo passo deste paradigma é a conquista, na qual ao obtermos o problema em sua menor escala devido a divisão realiza-se a computação deste subproblema. Por fim, a terceira fase é a combinação, na qual com as soluções dos subproblemas encontrados devem-se juntar para definir a solução macro do problema original.

Para além disso, pensando na aplicação deste paradigma no problema do Subvetor de Soma Máxima(SSM) definimos alguns parâmetros que devem ser calculados durante a execução do problema. O primeiro deles é a própria SSM, contudo para obtermos este valor precisaremos de calcular durante a execução de cada recursão o valor do prefixo de soma máxima, o sufixo de soma máxima e a soma total do subvetor. Além disso são calculados também o índice de início e fim do subvetor.

4. O algoritmo

O algoritmo do Subvetor de Soma Máxima, como dito antes, será implementado através da divisão e conquista. Para isso primeiro define-se o processo de divisão recursiva e o tamanho do menor subproblema desejado. O processo de divisão acontecerá sempre dividindo recursivamente o vetor em dois vetores menores, que comumente são chamados de vetor da direita(VD) e vetor da esquerda(VE), isso ocorre até chegar-se no menor subproblema. Já o tamanho do menor subproblema será definido quando o início e o fim deste vetor são iguais, ou seja, o vetor possui apenas um elemento.

Na fase de conquista ao chegarmos no vetor de tamanho um, que seria o nosso menor subproblema, o que é computado são os dados apresentados no tópico anterior, sendo a SSM, o sufixo, o prefixo e a soma iguais ao valor do único elemento do vetor e os índices de fim e início são os índices de fim e início do próprio vetor de elemento único.

Por fim, combina-se os subproblemas retornados a cada processo recursivo. Para isso, recalcula-se os parâmetros pré-estabelecidos com base no resultado

obtido pela chamada recursiva das duas partes do problema. Assim tem-se que a soma do subvetor é a soma de VD e VE; a SSM é o máximo entre a SSM de VE, a SSM de VD e a soma do sufixo de VD com o prefixo de VE; o sufixo é dado pelo máximo entre o sufixo de VD e a soma do sufixo de VE com a soma de VD; o prefixo é máximo entre o prefixo de VE e a soma do prefixo de VE com a soma de VD e o fim e início são dados através do valor da SSM, caso a SSM seja igual a SSM de VE eles receberam o início e fim de VE, já se a SSM for igual a SSM da direita eles receberam o início e fim de VD e por fim caso a SSM seja igual a soma do sufixo de VD com o prefixo de VE então ele recebe o início de VE e o fim de VD.

5. Implementação

A implementação completa deste problema foi feita em um arquivo .cpp e para o desenvolvimento do algoritmo a estrutura de dado padrão do c++ utilizada foi apenas o vector. Além disso foi criada uma struct chamada parametrosSSM que tem como membros os pontos flutuantes para SSM, soma, sufixo e prefixo e dois inteiros para os índices de início e fim.

O primeiro passo do programa é capturar as notas dadas pelos amigos para os shows, estas notas são somadas e salvas em um `std::vector` em que cada posição do vetor representa um dos shows. Importante ressaltar que foi adicionado na posição zero do vetor o valor -6000, apenas para manter a concordância com o problema estabelecido em que os shows são numerados iniciando do 1, assim ao colocar -6000 na posição zero ela nunca fará parte de algum subvetor de soma máxima, ou seja, será uma posição descartada.

Com o vetor montando chama-se o algoritmo do Subvetor de Soma Máxima que realiza os passos descritos nos tópicos anteriores e retorna um `parametrosSSM` como resposta. Assim sendo basta imprimir o início e o fim do `parametrosSSM` e teremos o intervalo de shows que estes amigos deverão assistir.

6. Conclusão

Por fim é importante ressaltar que o algoritmo implementado para a solução do Subvetor de Soma Máxima possui complexidade temporal $O(n*m)$, no qual n é o número de shows e m o número de amigos. Esta complexidade é resultado da análise da entrada no qual precisa-se receber e somar cada nota dada pelos m amigos. Tirando a captura da entrada o algoritmo de divisão em conquista implementado para resolver o problema do Subvetor de Soma Máxima tem complexidade $O(n \log n)$, pois devido ao seu processo de dividir o vetor original recursivamente são gerados $\log n$ níveis de recursão e no último nível serão necessárias n “conquistas” para a resolução do problema.