

Thomas Santos Pollarini
Samuel Rodrigues Viana de Faria

Relatório sobre o Algoritmo para o Cálculo do Maior Subarray Contíguo usando a Divisão e Conquista

Tema Abordado: O código apresentado implementa um algoritmo para encontrar o maior subarray contíguo em um array dado. O problema do Maior Subarray Contíguo tem como objetivo determinar a soma máxima que pode ser obtida escolhendo uma subsequência contígua de elementos em um array.

Funções Utilizadas:

1. **calculaSoma(subArray):** Esta função calcula a soma dos elementos de um subarray passado como argumento. Ela percorre o subarray e acumula os valores em uma variável, retornando a soma.
2. **maiorSubArray(array, posEsq, posDir):** Esta é a função principal que implementa o algoritmo de divisão e conquista para encontrar o maior subarray contíguo. Ela recebe o array a ser processado e os índices de início (posEsq) e fim (posDir) do array.

Modelagem do Problema:

O problema do Maior Subarray Contíguo é modelado como um problema de dividir e conquistar. A ideia básica é dividir o array em duas metades, encontrar o maior subarray em cada metade e, em seguida, encontrar o maior subarray que cruza o ponto médio. O subarray que cruza o ponto médio é importante, pois pode ser o maior subarray global.

Solução do Problema utilizando Divisão e Conquista:

O algoritmo funciona da seguinte forma:

1. **Caso Base:** Se houver apenas um elemento no array (quando posEsq é igual a posDir), o maior subarray é esse único elemento, e a função retorna esse valor.
2. **Divisão do Array:** O array é dividido em duas metades, uma à esquerda e outra à direita, calculando o ponto médio (meioArray).
3. **Recursão:** As funções maiorSubArray são chamadas recursivamente para encontrar o maior subarray nas metades esquerda e direita do array.
4. **Cruzamento do Meio:** A função procura o maior subarray que cruza o ponto médio do array, ou seja, um subarray que contém elementos de ambas as metades.
5. **Comparação:** O algoritmo compara os resultados das três opções (maior subarray na metade esquerda, maior subarray na metade direita e maior subarray cruzando o meio) e retorna o maior deles.

Execução do Código:

O código cria um array com tamanho e valores definidos pelo usuário, em seguida, encontra o maior subarray contíguo e o imprime na tela. A maior soma de acordo com o subarray correspondente e o mesmo são exibidos no final da execução.

Conclusão:

O algoritmo para calcular o maior subarray contíguo usando a técnica de divisão e conquista é uma solução eficaz para esse problema, por outro lado é uma técnica complexa de ser programada. Ele divide o problema em subproblemas menores, resolve esses subproblemas e combina suas soluções para obter o resultado final. Esse algoritmo é um exemplo de como a estratégia de divisão e conquista pode ser aplicada para resolver problemas complexos de maneira eficiente.