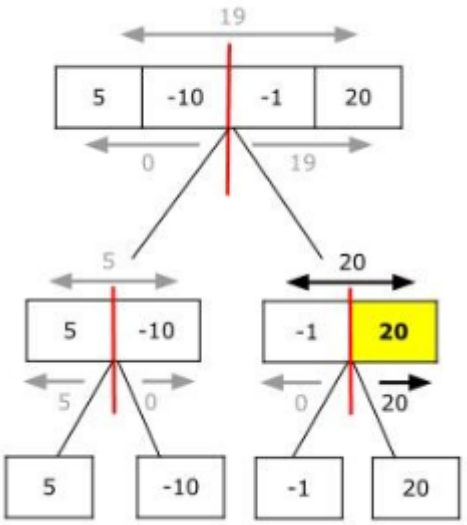
Trabalho 3: Algoritmo para o Cálculo do Maior Subarray Contíguo, com Divisão e Conquista

Nomes: Henrique Fugie de Macedo e Luiz Otávio Silva Cunha
Matrículas: 0056151 e 0056153

Algoritmo para o Cálculo do Maior Subarray Contíguo

O problema do Maior Subarray Contíguo, também conhecido como "Maximum Subarray Sum" ou "Máxima Soma de Subsequência Contígua", consiste em encontrar a soma máxima de um subarray (sequência de elementos consecutivos) em um array dado. Este é um problema clássico de programação dinâmica e pode ser resolvido de forma eficiente utilizando a técnica de Kadane. O algoritmo de Kadane mantém a soma máxima atual e atualiza-a conforme percorre o array, ignorando subarrays cuja soma se torna negativa.



O algoritmo de Kadane aproveita o fato de que, se a soma de um subarray se torna negativa, não é benéfico incluir esse subarray em um subarray maior, pois isso diminuiria a soma total. Portanto, o algoritmo "abandona" subarrays negativos, começando a calcular a soma a partir do próximo elemento do array.

O tempo de execução do algoritmo de Kadane é linear, O(n), tornando-o uma escolha eficiente para resolver o problema do Maior Subarray Contíguo. É uma abordagem simples e elegante que não requer armazenamento adicional de subarrays intermediários, economizando espaço e melhorando a eficiência.

Divisão e Conquista

A abordagem de divisão e conquista é uma técnica algorítmica fundamental que consiste em resolver um problema dividindo-o em subproblemas menores, resolvendo esses subproblemas de forma recursiva e, em seguida, combinando suas soluções para obter a solução do problema original. Vamos discutir como essa abordagem funciona, seus pontos positivos e alguns exemplos de problemas em que é comumente aplicada.

O funcionamento dela ocorre da seguinte forma: ele começa dividindo o problema original em subproblemas menores e mais simples, em seguida cada subproblema é resolvido de forma recursiva e no final os resultados são combinados para formar a solução do problema original.

Explicação do código

O código foi feito utilizando divisão e conquista, onde ele começa executando um Menu que possibilita o usuário escolher se quer gerar um array com valores aleatórios de 10 posições ou se quer enviar um array digitado por ele mesmo. Depois da apresentação deste menu, o programa chama a função principal que começa verificando o caso base do problema, que seria o início igual ao fim, e faz uma chamada recursiva que resulta na parte da esquerda do programa, e ele vai fazendo isso novamente com o vetor que ele receber, o lado esquerdo, até encontrar o caso base, que quando é executado, retorna a primeira posição do array, e chama recursivamente de novo, porém agora para resolver a posição da direita, que neste caso, também vai atingir o caso base na primeira execução, em seguida, é chamada uma função que faz a verificação da soma que seria basicamente a junção do que foi encontrado na esquerda com o que foi encontrado na direita, e depois disso tudo, ele vai executar uma verificação para identificar, dentre os 3 resultados, qual deles possui o maior valor de soma, que é o que o problema quer, porém não neste momento, pois para ele receber o resultado final, ele deve resolver toda a recursão do problema ,até que ele alcance o primeiro caso, que foi a divisão do array ao meio, esquerda e direita,que ao a cruzada seria analisando o array completo, ai deste forma ele retornaria o valor correto. Aqui está uma imagem exemplificando isso, onde a da esquerda mostra um momento que está sendo resolvido uma chamada recursiva no meio do caminho e o da direita o resultado final.