

Lista de Exercícios 3

Universidade Federal do Ceará - Campus Quixadá
Projeto e Análise de Algoritmo — QXD0041 – 2023.2
Prof. Fabio Dias

Divisão e Conquista

1. **Problema do Teto em um Vetor:** Dado um vetor ordenado de tamanho n e um inteiro k , encontrar o teto de k dentro do vetor. O teto de um número k no vetor, é o menor número do vetor que é maior ou igual k . Mostre um algoritmo de Divisão e Conquista que resolve esse problema com tempo $O(\log n)$.
2. **Encontrar o número de 1's em um vetor binário ordenado:** Dado um vetor binário (contendo apenas 0's e 1's) ordenado, encontre a quantidade de números 1's nesse vetor. Mostre um algoritmo de Divisão e Conquista que resolve esse problema com tempo $O(\log n)$.
3. Vamos supor uma versão diferente do Merge-Sort, onde ele divide o vetor em 3 sub-vetores de tamanhos aproximadamente iguais. Depois ordena os 3 sub-vetores recursivamente e utiliza uma versão alterada do Intercala para realizar o Merge desses 3 subvetores ordenados em um único ordenado. Mostre o algoritmo dessa versão do Merge-Sort junto com o Intercala alterado. Realize a análise de complexidade desse algoritmo e explique se essa versão é mais eficiente do que a versão original.
4. **Problema do k -ésimo mínimo:** Dado um vetor de n elementos e um inteiro $k \leq n$, encontrar o elemento do vetor que seja o k -ésimo menor elemento do vetor, ou seja, o elemento do vetor que estaria na posição k se o vetor estivesse ordenado. Uma forma de resolver esse problema seria ordenar o vetor, usando algum método de ordenação, e depois retornar o elemento da posição k . Desenvolva um algoritmo de Divisão e Conquista para resolver esse problema sem a necessidade de ordenar o vetor. Faça a análise de complexidade desse algoritmo e compare com a versão que ordena o vetor.
5. Considere o seguinte problema: dados n intervalos na reta real, $I_i = [a_i, b_i]$, $i = 1, 2, \dots, n$, desenvolva um algoritmo que lista todos os intervalos que estão contidos dentro de pelo menos um dos outros intervalos passados na entrada. O seu algoritmo deve ter complexidade $O(n \log n)$.
6. Dado dois números inteiros a e n , precisamos calcular a^n executando $O(\log n)$ operações de multiplicação. Projete um algoritmo de divisão e conquista para calcular a^n com complexidade de $O(\log n)$ operações de multiplicação.
7. Um vetor de tamanho n possui a seguinte propriedade: os elementos de 1 até o índice p estão ordenados em ordem crescente, e os elementos de p até n em ordem decrescente. Assumindo que todos os valores são distintos, desenvolva um algoritmo $O(\log n)$ que determina o índice p .
8. Uma nova companhia aérea de passageiros que atenderá n cidades do sertão central do Ceará, precisa planejar os seus voos. Para permitir que todos os passageiros possam viajar de qualquer cidade para qualquer outra cidade com um voo direto,

seria necessários $\Omega(n^2)$ rotas diferentes. A companhia aérea não pode ser lucrativa ao suportar tantas rotas. Outra opção seria ordenar (em relação a um sentido, exemplo, de norte a sul) as cidades em uma lista, e ter voos que vão da cidade no índice i , para a cidade no índice $i + 1$. Isso, no entanto, significaria que alguns passageiros precisariam de $\Omega(n)$ conexões para chegar ao seu destino (passageiros com origem na cidade 1 e destino na cidade n). É necessário elaborar um conjunto de rotas que não exija que nenhum passageiro tenha mais do que uma única conexão (ou seja, deve tomar no máximo dois voos) e não exija mais do que $O(n \log n)$ rotas. Projete um algoritmo de divisão e conquista que devolva esse conjunto de rotas com as exigências dadas.

9. Suponha que esteja escolhendo entre os seguintes três algoritmos:

- Algoritmo A resolve problemas de tamanho n dividindo-os em cinco subproblemas de metade do tamanho, solucionando cada subproblema recursivamente e, então, combinando as soluções em tempo linear.
- Algoritmo B resolve problemas de tamanho n resolvendo recursivamente dois subproblemas de tamanho $(n - 1)$ e, então, combinando as soluções em tempo constante.
- Algoritmo C soluciona problemas de tamanho n dividindo-os em nove subproblemas de tamanho $n/3$, resolvendo recursivamente cada subproblema e, então, combinando as respostas em tempo $O(n^2)$.

Qual o tempo de execução de cada um desses algoritmos (em notação O ou Θ) e qual você escolheria?