Universidade Federal do Ceará

Campus de Quixadá

QXD0041 - Projeto e Análise de Algoritmos

Lista 3 - Algoritmo Guloso

1. Projete um algoritmo para encontrar os k menores elementos de um conjunto não ordenados de n inteiros em $\mathcal{O}(n + k\ logn)$.

2. Projete um algoritmo eficiente para calcular a união dos conjuntos A e B, onde $n = max(|A|, |B|)$. A saída deve ser um vetor de elementos distintos que formam a união dos conjuntos.

   (a) Assuma que A e B não são selecionados. Dê um algoritmo $\mathcal{O}(n\ logn)$ para o problema.

   (b) Suponha que A e B sejam classificados. Dê um algoritmo $\mathcal{O}(n)$ para o problema.

3. Para cada um dos seguintes problemas, dê um algoritmo que encontre os números desejados com a complexidade dada. Para manter suas respostas resumidas, sinta-se à vontade para usar algoritmos de ordenação conhecidos como subrotinas. Para o exemplo, $S = 6, 13, 19, 3, 8$, $19 - 3$ maximiza a diferença, enquanto $8 - 6$ minimizam a diferença.

   (a) Seja S uma vetor não ordenada de n inteiros. Dê um algoritmo que encontre o par $x, y \in S$ que maximiza $|x - y|$. Seu algoritmo deve ser executado em $\mathcal{O}(n)$ o pior caso.

   (b) Seja S uma matriz ordenada de n inteiros. Dê um algoritmo que encontre o par $x, y \in S$ que maximiza $|x - y|$. Seu algoritmo deve ser executado em $O(1)$ pior caso.

4. Dada uma sequência de $x_1, x_2, \ldots, x_{2n}$. Projete um algoritmo $O(n\ logn)$ que particione a entrada em 2 conjuntos de n elementos. Para cada conjunto, computamos a soma de seus números. Denote por s_1 e s_2 as somas das duas partições. O seu algoritmo deve encontrar uma partição que maximiza a diferença entre $s1$ e $s2$.

5. Mostre que o seu algoritmo está correto.

6. A entrada é uma sequência de números $x_1, x_2, \ldots, x_n$ onde n é par. Projete um algoritmo que particione a entrada em $n/2$ pares da seguinte maneira. Para cada par, computamos a soma de seus números. Denote por $s_1, s_2, \ldots, s_{n/2}$ as $n/2$ somas. O algoritmo deve encontrar uma partição que minimize a máximo das somas e deve ser tão eficiente quanto possível. Por exemplo, dados os números (1,3,5,9). As possíveis partições são {(1,3),(5,9)},{(1,5),(3,9)}, {(1,9),(3,5)}. A soma dos pares para essas partições são (4,14),(6,12) e (10,8). Assim a terceira partição tem como soma máxima 10 que é o mínimo entre as três partições possíveis. Explique porque ele funciona e determine a sua complexidade.

7. Mostre que seu algoritmo tem a propriedade gulosa, ou seja, existe uma solução ótima que contém a primeira escolha gulosa do algoritmo.

8. Suponha dado um conjunto de livros numerados de 1 a n. Suponha que o livro i tem peso $p[i]$ e que $0 < p[i] < 1$ para cada i. Considere o problema de acondicionar os livros no menor número possível de envelopes de modo que cada envelope tenha no máximo 2 livros e o peso do conteúdo de cada envelope seja no máximo 1. Escreva um algoritmo guloso que recebe um vetor $p[1..n]$ e devolve o número mínimo de envelopes. O consumo de tempo do seu algoritmo deve ser $\mathcal{O}(n\ lgn)$.

9. Suponha que a é um livro de peso máximo e b é um livro de peso mínimo.

   (a) Se $p[a] + p[b] > 1$, então $\{a\}$ é um envelope em todo acondicionamento.

   (b) Se $p[a] + p[b] \leq 1$, então existe acondicionamento mínimo que possui $\{a, b\}$