Projeto de algoritmos recursivos e divisão e conquista

Projeto de algoritmos recursivos

Questão 1. (Manber) O quebra-cabeça das torres de Hanoi é um exemplo de um problem não trival que tem uma solução simples recursiva. Há n discos colocado em uma estaca em ordem decrescente de tamanho. Há duas estacas livres. O objetivo do quebra-cabeças é mover todos os discos, um por vez, da primeira estaca até outra estaca da seguinte maneira. Discos são movidos do topo de uma estaca para o topo de outra. Um disco só pode ser movido se for menor do que todos os outros discos na estaca de destino. Em outras palavas, a ordenação dos discos em ordem decrescente deve ser mantida em todos os momentos. O objetivo é mover todos os discos com o menor número de movimentos.

(a) Projete um algoritmo (por indução) para encontra uma sequência mínima de movimentos que resolve as torres de Hanoi para o problema com n discos.

(b) Quantos movimentos são realizados pelo seu algoritmo? Construa uma relação de recorrência para o número de movimentos e resolva-a (exatamente).

(c) Mostre que o número de movimentos da parte (b) é ótimo, i.e., mostre que não pode existir outro algoritmo que realiza menos movimentos.

Questão 2. (Manber) O seguinte é uma variação do problema das torres de Hanoi. Não assumimos mais que os discos estão em uma única estaca. Eles podem estar distribuídos entre as três estacas, desde que estejam ordenados em cada uma. O propósito dessa variante continua sendo mover todos os discos para uma estaca especificada, com as mesmas restrições do problema original, com tão poucos movimentos quanto possível. Projete um algoritmo para encontrar uma menor sequência de movimentos para essa versão das torres de Hanoi para n discos.

Questão 3. Considere o problema da subsequência consecutiva par máxima em que, dado um vetor, queremos encontrar uma subsequência (de números consecutivos) cuja soma seja um número par com o maior valor. Projete um algoritmo usando indução para esse problema. Analise o seu tempo de execução usando uma recorrência.

Questão 4. (Manber) Seja $x_1, x_2, \ldots, x_n$ uma sequência de números reais (não necessariamente positivos). Projete um algoritmo $O(n)$ para encontrar a subsequência $x_i, x_{i+1}, \ldots, x_j$ (de elementos consecutivos) de tal forma que o produto dos números é máximo entre todas as subsequências consecutivas. O produto de uma subsequência vazia é definido como 1.

Questão 5. Sobre os problemas a seguir, marque certo ou errado:

(a) C E É possível calcular o valor de um polinômio de quarto grau calculando o valor de um outro polinômio de terceiro grau relacionado.

(b) C E Existe algoritmo para calcular o valor de um polinômio $P(x) = a_3x^3 + a_2x^2 + a_1x + a_0$ que faz menos de $3+2+1 = 6$ multiplicações.

(c) C E Em um algoritmo recursivo, fortalecer a hipótese da indução significa restringir o conjunto de instâncias que o algoritmo resolve.

(d) C E Em um algoritmo recursivo projetado por indução, a solução de um problema de tamanho $n+1$ deve ser obtida a partir da solução do problema de tamanho n.

(e) C E Defina como *impopular* em um grupo de pessoas, alguém que conhece todos no grupo, mas que não é conhecida por mais ninguém. Sempre há no máximo uma impopular no grupo.

(f) C E Defina como *celebridade* em um grupo de pessoas, alguém que é conhecida por todos no grupo, mas que não conhece ninguém. Sempre há pelo menos uma celebridades no grupo.

(g) C E Se A *conhece* B ou B *não conhece* A, então A não é uma celebridade.

(h) C E Na sequência $(2,-3,1,3,-2,1,-1,4,-4,1,2)$, uma subsequência **consecutiva** de soma máxima é $(1,3,-2,1,-1,4)$.

(i) C E Na sequência $(2,-3,1,3,-2,1,-1,4,-4,1,2)$ uma subsequência **crescente** mais longa é $(2,1,3,1,4,1,2)$.

(j) C E Se quisermos encontrar os dois maiores números de um grupo de n elementos, devemos primeiro buscar o maior entre os n números e depois buscar o maior entre os $n-1$ números restantes, totalizando pelo menos $n-1+n-2$ comparações.

Divisão e conquista

Questão 6. Considere o problema de encontrar o máximo em um vetor de n números. Projete um algoritmo de divisão e conquista para o problema. Experimente duas abordagens: usar um subproblema de tamanho $n-1$ e dividir o subproblema em dois subproblemas de tamanho aproximadamente $n/2$. Qual a melhor abordagem?

Questão 7. (Kleinberg e Tardos, Solved Exercise 2) Suponha que você está prestando consultoria em uma empresa de investimentos e gostaria de saber os melhores dias de comprar uma ação e vendê-la posteriormente, isso é, em que dia se deve comprar e em que dia se deve vendê-la para maximizar o lucro? Suponha que você tenha a estimativa dos preços de n dias. Escreva um algoritmo que execute em tempo $O(n\log n)$ baseando-se no princípio da divisão e conquista.

Questão 8. (Baseado em Solved Exercise 1 de Kleinberg e Tardos) Suponha que você tenha um vetor A de n números naturais. Suponha que esse vetor tem a seguinte propriedade:

- se o máximo ocorre em $A[p]$ para um índice p, então $A[1] \leq A[2] \leq \ldots A[p]$ e $A[p] \geq A[p+1] \geq \ldots A[n]$;
- se $A[i] = A[j] = x$, então $A[k] = x$ sempre que $i \leq k \leq j$.

Seu objetivo é encontrar o valor máximo nesse vetor. Escreva o algoritmo mais eficiente que conseguir. Justifique a complexidade e a correção.

Questão 9. Leia a seção sobre o algoritmo de Strassen de CLRS (seção 4.2 na 3ª Edição e seção 28.2 na 2ª Edição).