

## Projeto de algoritmos por indução

**Questão 1.** (Manber) O quebra-cabeça das torres de Hanoi é um exemplo de um problema não trivial que tem uma solução simples recursiva. Há  $n$  discos colocados 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 palavras, 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 encontrar 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 realize 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.

## Divisão e conquista

**Questão 4.** 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 5.** (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 6.** Leia a seção sobre o algoritmo de Strassen de CLRS (seção 4.2 na 3ª Edição e seção 28.2 na 2ª Edição).

---

<sup>1</sup>Esta lista deve ser feita logo após as aulas do conteúdo correspondente e serve para fixar o conteúdo, confirmar ou identificar as dúvidas. Anote suas dúvidas e procure atendimento! Os exercícios são referências ou transcrições de exercícios dos livros-textos (CLRS/Manber), ou foram gentilmente cedidos por outros professores, particularmente por Flávio Keidi Miyazawa (FKM), Cid Carvalho de Souza e Orlando Lee (CID/OL).