

Lista 1 – Projeto e Análise de Algoritmos – 2024/1

Programa de Pós-Graduação em Ciência da Computação – Universidade Federal de Santa Catarina
Prof. Alexandre Goncalves Silva e Prof. Rafael de Santiago

Observações gerais:

- Trabalho deve ser executado em, no máximo, 3 estudantes da disciplina;
- Um documento com as respostas deve ser preparado e entregue em formato PDF;
- O nome dos membros da equipe deve constar no documento entregue;
- Apenas um membro de cada equipe deverá realizar a entrega;
- A entrega deverá ser realizada no MOODLE.

1. (1 ponto) Prove as seguintes séries por indução matemática:

(a) $\sum_{i=0}^n i^2 = \frac{n(n+1)(2n+1)}{6}$, $n \geq 1$

(b) $\sum_{i=0}^n i^3 = \frac{n^2(n+1)^2}{4}$, $n \geq 1$

2. (2 pontos) Resolva as recorrências por meio do **método de iteração** ou **expansão telescópica**, e faça a prova, por indução, da fórmula fechada:

(a) $T(1) = k$; $T(n) = cT(n-1)$; c, k constantes e $n > 0$

(b) $T(1) = 1$; $T(n) = 3T(n/4) + n^2$

3. (1 ponto) Use o **teorema mestre** para determinar o custo de execução de seus algoritmos expressos pelas seguintes recorrências:

(a) $T(n) = 3T(n/2) + n \log n$

(b) $T(n) = 64T(n/4) + n^2$

4. (2 pontos) Use o **método de árvore de recursão** para determinar o custo de execução dos algoritmos expressos pelas recorrências abaixo:

(a) $T(1) = 1$; $T(n) = 2T(n/2) + \Theta(n^2)$

(b) $T(1) = 1$; $T(n) = 2T(n/3) + \Theta(n)$

5. (2 pontos) Suponha que, para entradas de tamanho n , você tenha que escolher um dentre três algoritmos A , B e C .

(a) Algoritmo A resolve problemas dividindo-os em oito subproblemas de tamanho $n/4$, recursivamente resolve cada subproblema e então combina as soluções em tempo $O(1)$.

(b) Algoritmo B resolve problemas dividindo-os em quatro subproblemas de metade do tamanho, recursivamente resolve cada subproblema e então combina as soluções em tempo $O(n)$.

(c) Algoritmo C resolve problemas dividindo-os em dois subproblemas de tamanho $n-1$, recursivamente resolve cada subproblema e então combina as soluções em tempo $O(n^2)$.

Qual o consumo de cada um desses algoritmos? Expresse as suas respostas em termos da notação O , mas procure dar as respostas com funções para limites superiores mais próximos possíveis. Qual algoritmo é assintoticamente mais eficiente no pior caso? Justifique as suas respostas.

6. (**2 pontos**) Suponha que você foi contratado para prestar uma consultoria a um banco que está preocupado com a detecção de fraudes em cartões. Eles tem um conjunto de n cartões que foram confiscados por suspeita de fraude. Cada cartão de banco é um pequeno objeto de plástico contendo um conteúdo encriptado que corresponde a uma conta única no banco. Cada conta pode ter muitos cartões associados, e diz-se que dois cartões são equivalentes se os mesmos correspondem a mesma conta. É muito difícil ler um número de conta do banco diretamente, mas o mesmo disponibilizou um “testador de equivalência” que recebe dois cartões e determina se eles são equivalentes (resposta booleana, ou seja, igual ou diferente; não é possível a verificação de ordem, por exemplo, menor ou maior). O que se deseja saber é: há algum cartão se repetindo ao menos 50% dos n cartões disponíveis? Para isso, crie um pseudo-código para um algoritmo de divisão e conquista que resolva o problema, respondendo a questão, e demonstre que sua solução realiza apenas $O(n \log n)$ invocações do “testador de equivalências”.