

 <p>Universidade Federal do Ceará Centro de Ciências Departamento de Computação</p>	<p>2ª Avaliação Parcial Construção e Análise de Algoritmos (ck0183/ck0203) - 2021.1 Profa. Ana Karolinna Maia karolmaia@ufc.br</p> <p>Aluno:  Matrícula:</p>	<p>Nota</p>
--	--	-------------

- A prova deve ser escrita a mão (no papel) e fotografada/escaneada para enviar para correção em um único arquivo no formato PDF.
- O upload do arquivo deve ser feito pelo SIGAA.
- Respostas em linguagens de programação não serão aceitas.

**Questão 1** (2,0 pontos). No problema da pilha de caixas são dadas  $n$  caixas  $C_1, C_2, \dots, C_n$ . Cada caixa  $C_i$  possui peso  $p_i$  e quantidade de objetos  $k_i$ . O problema consiste em construir uma pilha  $P$  com todas as caixas, de modo a minimizar o esforço, calculado como  $esforo(P) = \sum_{j=1}^n (k_j \times (\sum_{i=1}^j p_i))$ . Prove ou apresente contra-exemplos para as seguintes ideias de como resolver o problema de forma gulosa:

- Ordenar as caixas de maneira crescente pelo peso;
- Ordenar as caixas de maneira decrescente pela quantidade de objetos.

**Questão 2** (4,0 pontos). Seja  $1, \dots, n$  um conjunto de tarefas. Em cada dia, é possível executar no máximo **OITO** tarefas, uma para cada hora de trabalho do dia. Os dias de trabalho são numerados como  $1, 2, 3, \dots$  e as horas de trabalho são numeradas de 1 a 8. Cada tarefa  $T$  tem um prazo  $P_T$ : a tarefa deveria ser executada em algum dia do intervalo  $1, \dots, P_T$ . Cada tarefa  $T$  tem uma multa  $M_T > 0$ : se uma tarefa  $T$  é executada depois do prazo  $P_T$ , sou obrigado a pagar a multa  $M_T$  (mas a multa não depende do número de dias de atraso). Problema: Programar as tarefas informando em qual dia e hora cada tarefa deve ser executada de modo a minimizar a multa total. Descreva com palavras um algoritmo **guloso** para resolver o problema. Argumente porque seu algoritmo está correto.

**Questão 3** (4,0 pontos). Você deve cortar um tronco de madeira em vários pedaços. A empresa mais em conta para fazer isso é a União Fácil Corte (UFC), que cobra de acordo com o comprimento do tronco a ser cortado. A máquina de corte deles permite que apenas um corte seja feito por vez. Se queremos fazer vários cortes, é fácil ver que ordens diferentes destes cortes levam a preços diferentes. Por exemplo, considere uma tora com 10 metros de comprimento, que tem que ser cortada a 2, 4 e 7 metros de uma de suas extremidades. Há várias possibilidades. Podemos primeiramente fazer o corte dos 2 metros, depois dos 4 e depois dos 7. Tal ordem custa  $10+8+6 = 24$ , porque a primeira tora tinha comprimento 10, o que restou tinha 8 metros de comprimento e o último pedaço tinha comprimento 6. Se cortássemos na ordem 4, depois 2, depois 7, pagaríamos  $10 + 4 + 6 = 20$ , que é mais barato. Seu chefe encomendou um algoritmo de programação dinâmica que, dado o comprimento  $L$  do tronco e  $k$  pontos  $p_1, \dots, p_k$  de corte, encontre o custo mínimo para executar esses cortes na UFC.

- Explique a propriedade de subestrutura ótima desse problema.
- Escreva a equação de recorrência.
- Escreva o algoritmo em pseudo-código.