Universidade Federal do Ceará
Construção e Análise de Algoritmos
Prova 2

1. [2,5 pontos] Seja $P : \mathbb{N} \to \mathbb{N}$ uma função tal que: $P(n) = n$ para $n = 0, 1, \ldots, 20$ e, para $n \geq 21$,

$$P(n) = 3n + \sum_{k=0}^{10} P\Big(\Big\lfloor \frac{n}{2} \Big\rfloor + k \Big).$$

(a) Escreva um algoritmo recursivo puro que recebe um número n como entrada e retorna o valor exato de $P(n)$. Calcule a complexidade do seu algoritmo.

(b) Escreva um algoritmo de programação dinâmica para o mesmo problema e calcule a complexidade.

(c) Escreva um algoritmo de memoização (recursivo usando memória) e calcule a complexidade.

Compare os tempos dos três algoritmos.

2. [2,5 pontos] Seja $1, \ldots, n$ 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 $1, 2, 3, \ldots$ 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 $1, \ldots, P_T$. 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. Escreva um algoritmo **guloso** para resolver o problema. Argumente porque seu algoritmo está correto. Analise o consumo de tempo.

3. [2,5 pontos] Você recebe n números reais positivos $X = (x_1, x_2, \ldots, x_n)$ e uma sequência de $n - 1$ operadores $op = (op_1, \ldots, op_{n-1})$ em $\{+, -, \times\}$, onde $+$ é soma, $-$ é subtração e $\times$ é multiplicação. Essas sequências de números e operadores representam uma expressão matemática. Por exemplo, se $X = (0.3, 1, 4, 0.7, 0.2)$ e a sequência de operadores é $(+, \times, +, \times)$, então temos a expressão: $0.3 + 1 \times 4 + 0.7 \times 0.2$. Desejamos colocar parêntesis na expressão de modo que o resultado final seja o mínimo possivel. Também desejamos colocar parêntesis na expressão de modo que o resultado final seja o máximo possivel. Por exemplo, o máximo e o mínimo do exemplo são respectivamente $(0.3 + 1) \times (4 + (0.7 \times 0.2)) = 5.382$ e $(0.3 + (1 \times 4) + 0.7) \times 0.2 = 1$. Escreva um algoritmo de programação dinâmica que retorne o maior e o menor valor possível. A complexidade deve ser no máximo $O(n^3)$.

4. [2,5 pontos] Você recebe uma sequência $S[1 \ldots n]$ com n dígitos de 0 a 9 e deseja saber se é possível quebrá-la em EXATAMENTE SEIS **números primos**. Por exemplo, se $S = 23571113$, então a resposta é SIM, pois S pode ser quebrada da seguinte forma $2, 3, 5, 7, 11, 13$, que contém exatamente seis números primos. O seguinte modo $23, 571, 113$ contém apenas números primos, mas não é válido, pois não tem seis números. Escreva uma algoritmo de programação dinâmica que determina se sua sequência S satisfaz ou não esta condição. A complexidade deve ser no máximo $O(n^2)$. Obs: você não precisa fazer uma função para descobrir se um número é primo; você pode assumir que tal função já existe.

Dica: Não é obrigatório, mas ajuda muito a correção de sua prova se você explicar o algoritmo com suas palavras antes de escrever o pseudo-código).