

 <p>Universidade Federal do Ceará Centro de Ciências Departamento de Computação</p>	<p style="text-align: center;">Atividade Construção e Análise de Algoritmos (ck0203) Profa. Ana Karolinnia Maia karolmaia@ufc.br</p> <p>Aluno:</p> <p>Matrícula:</p>	<p style="text-align: center;">Nota</p>
--	---	---

Questão 1. Você recebe uma palavra com n caracteres $S[1 \dots n]$, que você pensa ser um texto corrompido no qual não há pontuação (por exemplo, “*euadoroprogramaçãodinâmica*”). Você deseja reconstruir o seu texto usando um dicionário que disponibiliza uma função booleana $dict(w)$ que retorna verdadeiro, se w é uma palavra do dicionário, e falso, caso contrário. Escreva um algoritmo que determina se seu texto pode ser reconstruído como uma sequência de palavras válidas. A complexidade deve ser no máximo $O(n^2)$, assumindo que a função $dict$ leva tempo constante. Caso seu texto seja válido, faça seu algoritmo escrever a sequência correta de palavras.

Questão 2. Suponha que você esteja gerenciando a construção de outdoors em uma autoestrada, um trecho de estrada muito movimentado que passa por M quilômetros. Os possíveis locais para outdoors são dados por números x_1, x_2, \dots, x_n , cada um no intervalo $[0, M]$ (especificando sua posição ao longo da rodovia, medida em quilômetros a partir do km 0). Se você colocar um outdoor em localização x_i , você recebe uma receita de $r_i > 0$. Os regulamentos impostos pelo polícia rodoviária exigem que nenhum dos outdoors deve estar a menos ou igual a 5 quilômetros um do outro. Você gostaria de colocar outdoors em um subconjunto dos pontos possíveis para maximizar seu receita total, sujeita a esta restrição. Forneça um algoritmo de programação dinâmica que tome uma instância desse problema como entrada e retorna a receita total máxima que pode ser obtidas. O tempo de execução do algoritmo deve ser polinomial.

Questão 3. Imagine que você comprou um livro em uma livraria por R\$ 69,24, que você pagou com R\$ 70 em dinheiro. Você tem 76 centavos de troco, e o caixa agora precisa decidir se lhe dará um punhado de 76 moedas de 1 centavo ou apenas quatro moedas ($25 + 25 + 25 + 1 = 76$). Dar troco neste exemplo é fácil, mas ele lança luz sobre um problema mais geral: como um caixa pode dar troco usando o menor número de moedas? Diferentes moedas têm diferentes valores possíveis, ou denominações. Nos Estados Unidos, as denominações das moedas são (100, 50, 25, 10, 5, 1); na República Romana, elas eram (120, 40, 30, 24, 20, 10, 5, 4, 1). Faça um algoritmo para calcular o menor número de moedas necessárias para dar o troco para um certo valor, dado um conjunto de moedas (ambos recebidos como entrada).

Questão 4. Uma pessoa vai votar pela 1ª vez e precisa tirar seu título de eleitor. Para isso, é preciso entregar vários documentos e fotos, assim como preencher vários formulários chatos. O TRE juntou vários atendentes no Castelão e colocou três fileiras de n mesas para o atendimento dos eleitores. Há uma fileira de mesas no meio do campo, uma fileira do lado direito e uma outra fileira de mesas do lado esquerdo do Castelão. Resumindo, temos as mesas $M_1^{(c)}, M_2^{(c)}, \dots, M_n^{(c)}$ no centro do campo, temos as mesas $M_1^{(d)}, M_2^{(d)}, \dots, M_n^{(d)}$ do lado direito e as mesas $M_1^{(e)}, M_2^{(e)}, \dots, M_n^{(e)}$ do lado esquerdo. Após sair da mesa k de uma fileira, o eleitor precisa ir para a mesa $k + 1$ de qualquer fileira (não necessariamente a mesma fileira). Alguns atendentes são mais rápidos do que os outros. O tempo gasto nas mesas $M_k^{(c)}$, $M_k^{(d)}$ e $M_k^{(e)}$ são respectivamente $t_k^{(c)}$, $t_k^{(d)}$ e $t_k^{(e)}$ em minutos ($k = 1, \dots, n$). A pessoa leva 5 minutos para ir do meio do campo até o lado direito ou esquerdo, pois precisa desviar de muitas pessoas. Para ir direto do lado esquerdo para o lado direito, ele leva 12 minutos, pois precisa desviar também das mesas do centro. Ela chegou às 13h da tarde e marcou um cinema no fim da tarde. Se conseguir sair do castelão antes das 17h, vai assistir *TRUQUE DE MESTRE - O 3 ATO*. Caso contrário, só haverá sessão para o *SOLDADO DE CHUMBO*. A pergunta é: ela vai conseguir chegar a tempo para assistir o filme bom, ou melhor desistir do cinema? Faça um algoritmo de programação dinâmica que responda isso e diga a sequência das mesas que a pessoa apressada deve seguir.

Questão 5. Uma balsa leva carros de um lado do rio para o outro. A balsa tem duas pistas para colocar os carros. Cada pista tem tamanho de L metros. Os carros que querem entrar na balsa estão em fila e devem ser colocados na ordem da fila. A fila tem n carros C_1, C_2, \dots, C_n com tamanhos T_1, T_2, \dots, T_n . Os tamanhos podem ser bastante diferentes. Queremos colocar o maior número de carros na balsa decidindo em qual faixa cada carro deve ser colocado. Elabore um algoritmo de programação dinâmica que resolva esse problema. Como dica, use uma matriz $M[k, A, B]$ que representa o maior número de carros que podem ser colocados na balsa considerando a fila de carros C_k, \dots, C_n , a pista 1 da balsa tendo comprimento de A metros e a pista 2 tendo B metros. Se descobrirmos o valor de $M[1, L, L]$ resolvemos a questão (explique rapidamente porque). (a) Escreva uma recursão para $M[k, A, B]$. (b) Escreva um algoritmo de programação dinâmica em pseudo-código para obter $M[1, L, L]$.