



UNIVERSIDADE  
FEDERAL DO CEARÁ  
CAMPUS DE CRATEÚS

Universidade Federal do Ceará - Campus Crateús  
Disciplina: Projeto e Análise de Algoritmos  
Professor: Rennan Dantas

Nome do Aluno(a):

Matrícula:

Lista para o trabalho 2

**Questão 01** Uma subsequência contígua de uma sequência  $S$  é uma subsequência de elementos consecutivos de  $S$ . Por exemplo, se  $S=(5\ 15\ -30\ 10\ -5\ 40\ 10)$ , então  $(15\ -30\ 10)$  é uma subsequência contígua de  $S$ , mas  $(5,15,40)$  não é. Escreva um algoritmo linear para a seguinte tarefa: receba como entrada uma sequência de números  $(a_1, a_2, \dots, a_n)$  e devolva a subsequência contígua cuja soma é máxima (um subsequência de tamanho zero tem soma zero). No exemplo anterior, a resposta seria a subsequência  $(10\ -5\ 40\ 10)$  cuja soma é 55. (Dica, para cada  $j \in \{1, 2, \dots, n\}$ , considere subsequências contíguas terminando exatamente na posição  $j$ ).

**Questão 02** Uma subsequência é palíndroma se ela é igual lendo da direita para esquerda ou lendo da esquerda para a direita. Por exemplo, a sequência  $(ACGTGTCAAATCG)$  possui muitas subsequências palíndromas, como  $(ACGCA)$  e  $(AGTGA)$ . Mas a subsequência  $(ACT)$  não é palíndroma. Escreva um algoritmo  $O(n^2)$  que recebe uma sequência  $S[1..n]$  e retorna a subsequência palíndroma de tamanho máximo.

**Questão 03** Você vai iniciar uma viagem bastante longa. Você inicia a viagem no Km 0 (zero). No seu percurso, existem  $n$  hotéis com quilometragens iguais a  $a_1 < a_2 < \dots < a_n$ , onde cada  $a_i$  é medido a partir do ponto de Km 0. Os únicos lugares onde você pode parar são esses hotéis, mas você não precisa parar em todos. Sua viagem termina no hotel do Km  $a_n$  que é o seu destino. Você idealmente gostaria de viajar 200km por dia, mas nem sempre isso é possível (depende do espaço entre os hotéis). Se você viajar menos de 200 km em um dia, seu pai reclama que quer chegar logo no destino final, mas se você viajar mais de 200km em um dia a sua mãe reclama que está cansada. Mais especificamente, se você viaja  $X$  km em um dia, você recebe  $(200 - X)^2$  reclamações. Você deseja planejar sua viagem de forma a minimizar o número de reclamações recebidos e manter a sua família em harmonia. Ou seja, minimizar o número total de reclamações recebidas em todos os dias viajados. Escreva um algoritmo que determina a sequência ótima de hotéis em que você deve parar.

**Questão 04** Você recebe uma sequência  $S[1..n]$  com  $n$  dígitos de 0 a 9 e deseja saber se é possível quebrá-la em números que sejam quadrados ou cubos perfeitos. Por exemplo, se  $S=125271448164$ , então a resposta é SIM, pois  $S$  pode ser quebrada da seguinte forma 125, 27, 144, 81, 64, cujos números são quadrados perfeitos ou cubos perfeitos ( $125=5^3$ ,  $27=3^3$ ,  $144=12^2$ ,  $81=9^2$ ,  $64=8^2$ ). Outra possibilidade seria: 1,25,27,144,8, 16,4. Escreva um algoritmo de programação dinâmica que determina se sua sequência  $S$  satisfaz ou não esta condição. Imprima a solução ótima. A complexidade deve ser no máximo  $O(n^2)$ .

**Questão 05** Escreva um algoritmo  $O(nT)$  que recebe um inteiro positivo  $T$  e uma lista de  $n$  inteiros positivos  $(a_1, a_2, \dots, a_n)$  e decide se existe algum subconjunto desses inteiros cuja soma é igual a  $T$ . (Dica: Observe subconjuntos  $(a_1, a_2, \dots, a_k)$  e verifique se a soma é  $s$  onde  $1 \leq k \leq n$  e  $1 \leq s \leq T$ ). Faça o rastreo da solução ótima e a apresente.

**Questão 06** São dados os valores  $x_1, x_2, \dots, x_n$  e você quer fazer uma troca de valor  $v$ , mas pode usar cada valor de moeda no máximo uma vez. Por exemplo, se os valores são 1, 5, 10, 20, então você pode fazer uma troca de  $16 = 1+15$  e de  $31=1+10+20$ , mas não de 40 (porque não pode usar 20 duas vezes).

Entrada: Inteiros positivos  $x_1, x_2, \dots, x_n$ ; outro inteiro  $v$ .

Saída: Você pode fazer uma troca de  $v$ , usando cada valor  $x_i$  no máximo uma vez?

Mostre como resolver este problema em tempo  $O(nv)$  usando programação dinâmica. Faça o rastreo da solução ótima e a apresente.

**Questão 07** Dado um suplemento ilimitado de moedas de valores  $x_1, x_2, \dots, x_n$ , desejamos trocar um valor  $V$  usando no máximo  $k$  moedas; isto é, queremos encontrar um conjunto com até  $k$  moedas cujo valor total é  $V$ . Isso pode não ser possível: por exemplo, se temos as moedas de valor 5 e 10 e  $k=6$ , então podemos trocar 55 mas não 65. **Forneça um algoritmo de programação dinâmica** para o seguinte problema.

Entrada:  $x_1, x_2, \dots, x_n; k; V$ .

Questão: É possível trocar o valor  $V$  usando no máximo  $k$  moedas com os valores  $x_1, x_2, \dots, x_n$ ?

**Faça o rastreo e imprima a solução ótima.**

**Questão 08** Você deve cortar uma tora 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 da tora a ser cortada. 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 desses 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 a 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 comprimento de 8 metros 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  $L$  da tora e  $k$  pontos  $p_1, \dots, p_k$  de corte da tora, encontre o custo mínimo para executar esses cortes na UFC.

**Questão 09** Uma balsa leva carros de um lado do rio para 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 pode 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) Explique sucintamente a propriedade de subestrutura ótima desse problema.

(b) Escreva uma recursão para  $M[k,A,B]$ . (c) Escreva um algoritmo de programação dinâmica para obter  $M[1,L,L]$ . (d) Altere o seu algoritmo para que ele diga em qual pista cada carro deve ser colocado.

**Questão 10** Escreva um algoritmo que obtenha o código de Huffman ternário, ou seja, um código de Huffman em que podemos codificar os símbolos com 0,1 e 2. Exemplifique esse algoritmo e também o código de Huffman normal para o seguinte exemplo: um texto com 100.000 caracteres onde os símbolos são A,B,C,D,E,F e G com frequências 40%, 15%, 11%, 10%, 12%, 7% e 5%. Quantos bits esse texto codificado terá tanto no caso normal como no ternário? Quantos bits esse texto teria se usássemos uma codificação de tamanho fixo? Compare cada algoritmo: melhorou ou piorou?

**Questão 11** Seja  $1, \dots, n$  um conjunto de tarefas. Cada tarefa consome um dia de trabalho; durante um dia somente uma das tarefas pode ser executada. Os dias são numerados de 1 a  $n$ . A cada tarefa  $T$  está associado um prazo  $P_T$ : a tarefa deve ser executada em algum dia do intervalo  $1, \dots, P_T$ . A cada tarefa  $T$  está associada uma multa  $M_T \geq 0$ . 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). Escreva um algoritmo guloso para programar as tarefas (ou seja, informar em qual dia cada tarefa será realizada) de modo a minimizar a multa total. Prove a corretude e analise o consumo de tempo.

**Questão 12** Um servidor tem  $n$  usuários esperando para serem servidos. O tempo de serviço requerido por usuário é conhecido previamente: é  $t_i$  minutos para o usuário  $i$ . Portanto se, por exemplo, os usuários são servidos em ordem crescente de  $i$ , então o  $i$ -ésimo usuário tem de esperar  $\sum_{j=1}^i t_j$  minutos.

Queremos minimizar o tempo total de espera

$$T = \sum_{i=1}^n (\text{tempo gasto pelo usuário } i \text{ na espera}).$$

Forneça um algoritmo guloso para calcular a ordem ótima na qual devemos processar os usuários. Explique resumidamente a propriedade da escolha gulosa.

**Questão 13** Quero dirigir um carro de uma cidade A a uma cidade B ao longo de uma rodovia. O tanque de combustível do carro tem capacidade suficiente para cobrir  $n$  quilômetros. O mapa da rodovia indica a localização dos postos de combustível. Escreva um algoritmo que garanta uma viagem com o número mínimo de reabastecimentos. Explique resumidamente a propriedade de escolha gulosa do seu algoritmo.

**Questão 14** Dado um vetor de  $n$  dígitos (cada dígito com valor de 0 a 9), encontre a menor soma possível de dois números formados por dígitos desse vetor. Todos os dígitos devem ser usados para formar os dois números. Cada dígito só pode ser usado uma vez. Esse problema tem a propriedade da escolha gulosa? Se sim, prove a propriedade da escolha gulosa e escreva um algoritmo guloso para resolvê-lo. Se não, elabore um algoritmo de programação dinâmica que resolva o problema.