Universidade Federal do Ceará
Campus Russas
Projeto e Análise de Algoritmos

Lista de exercícios 2

Programação Dinâmica

1. Seja $P : \mathbb{N} \to \mathbb{N}$ uma função definida da seguinte forma: $P(0) = P(1) = P(2) = P(3) = P(4) = 0$ e, para $n \geq 5$,

$$P(n) = P\Big(\lfloor \frac{n}{2} \rfloor\Big) \; + \; P\Big(\lfloor \frac{n}{2} \rfloor + 1\Big) \; + \; P\Big(\lfloor \frac{n}{2} \rfloor + 2\Big) \; + \; n.$$

(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 e calcule a complexidade.

2. 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 $a_1 < a_2 < \ldots < a_n$, onde cada a_i é medido a partir do ponto do Km 0. Os únicos lugares que você pode parar são esses hotéis, mas você não precisa parar em todos. Sua viagem termina no hotél do Km a_n que é o seu destino. Você idealmente gostaria de viajar 200 Km por dia, mas nem sempre isso é possível (depende do espaço entre os hotéis). Se você viaja menos de 200 Km em um dia, seu pai reclama que quer chegar logo ao destino final, mas se você viaja mais de 200 Km em um dia, 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 sua família em harmonia. Ou seja, minimizar o número total de reclamações recebidas em todos os dias viajados. Escreva um algoritmo de programação dinâmica que determina a sequência ótima de hotéis em que você deve parar.

3. 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 $(a_1, a_2, \ldots, a_n)$ e devolva a subsequência contígua cuja soma é máxima (uma 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 $j \in \{1, 2, \ldots, n\}$, considere subsequências contíguas terminando exatamente na posição j).

4. Você recebe uma sequência $S[1 \ldots n]$ 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 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. A complexidade deve ser no máximo $O(n^2)$. Caso a resposta seja SIM, faça seu algoritmo escrever a sequência correta de quadrados e/ou cubos perfeitos.

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 $C_1, C_2, \ldots, C_n$ com tamanhos $T_1, T_2, \ldots, T_n$. 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 $C_k, \ldots, C_n$, 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 seu algoritmo para que ele diga em qual pista cada carro deve ser colocado.

6. Escreva um algoritmo $O(nT)$ que recebe um inteiro positivo T e uma lista com n inteiros positivos $(a_1, a_2, \ldots, a_n)$ e decide se existe algum subconjunto cuja soma é igual a T. (Dica: Observe subconjuntos $(a_1, a_2, \ldots, a_k)$ e verifique se a soma é s onde $1 \leq k \leq n$ e $1 \leq s \leq T$).

Algoritmos Gulosos

7. Escreva um algoritmo que obtenha um código de Huffman ternário, ou seja, um código de Huffman em que podemos codificar os símbolos com 0, 1 e 2. Escreva agora um algoritmo que obtenha um código de Huffman ao-contrário, ou seja, obtenha o pior código livre de prefixo possível. Exemplifique esses dois algoritmos 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 nesses algoritmos? Quantos bits esse texto teria se usássemos uma codificação de tamanho fixo? Compare cada algoritmo: melhorou ou piorou?

8. Considere um conjunto de livros numerados de 1 a n. Suponha que o livro i tem peso p_i e que $0 < p_i < 1$ para cada i. Problema: Dado n e os números $p_1, \ldots, p_n$, acondicionar os livros no menor número possível de envelopes de modo que cada envelope tenha no máximo 2 livros e o peso do conteúdo de cada envelope seja no máximo 1. Escreva um algoritmo eficiente que resolva esse problema. Aplique seu algoritmo a um exemplo interessante. Mostre que seu algoritmo está correto.

9. Escreva um algoritmo eficiente que receba como entrada um conjunto de variáveis $x_1, \ldots, x_n$ e dois conjuntos I e D de pares (x_i, x_j) de variáveis. Os pares (x_i, x_j) em I representam restrições de igualdade $x_i = x_j$ e os pares (x_a, x_b) em D representam restrições de desigualdade $x_a \neq x_b$. Seu algoritmo deve responder se é possível ou não satisfazer todas as restrições em I e em D. Por exemplo, a seguinte entrada não é satisfatível: $I = \{(x_1, x_2), (x_2, x_3), (x_3, x_4)\}$ e $D = \{(x_1, x_4)\}$.

10. Um certo Fulano deseja fazer uma festa e está decidindo quem deve chamar. Ele tem n amigos para convidar e tem uma lista dos pares de amigos que se conhecem. Ele não quer que ninguém se sinta deslocado, mas também quer que a festa seja interessante e que pessoas que não se conhecem façam amizade. Assim ele se colocou as seguintes restrições: Para cada convidado, devem existir pelo menos dez pessoas na festa que ele conhece e dez pessoas na festa que ele não conhece. Faça um algoritmo que receba uma lista com n pessoas e uma lista com os pares dessas pessoas que se conhecem e devolva o maior número pessoas que poderão ser convidadas sob estas restrições. (a) Descreva com palavras como será o seu algoritmo. (b) Escreva o algoritmo em pseudo-código.