

Lista de exercícios 2 (Programação Dinâmica)
Cana - 2019.1

Questão 1. Dado um vetor $A[1 \dots n]$, dizemos que uma subsequência $S = [a_{i_1}, a_{i_2}, \dots, a_{i_k}]$ de A é densa se para todo $j \in 1, \dots, n$ (j representa uma posição do vetor A) temos que, ou $a_j \in S$, ou $a_{j-1} \in S$, ou $a_{j+1} \in S$. Escreva um algoritmo que recebe uma sequência A e encontra a subsequência densa de A cuja soma dos elementos é a menor possível.

Questão 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 $a_1 < a_2 < \dots < a_n$, onde cada a_i é medido a partir do ponto de Km 0. Os únicos lugares que 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 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, um de seus companheiros de viagem reclama que quer chegar logo ao destino final, mas se você viaja mais de 200 Km em um dia, o outro reclama que está cansado. 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 recebidas, 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 3. 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 4. Uma subsequência é palíndroma se ela é igual lendo da direita para esquerda ou lendo da esquerda para direita. Por exemplo, a sequência (ACGTGTCAAAATCG) possui muitas subsequências palíndromas, como (ACGCA) e (AGTGA). Mas a subsequência (ACT) não é palíndroma. Escreva um algoritmo em $O(n^2)$ que recebe uma sequência $S[1 \dots n]$ e retorna a subsequência palíndroma de tamanho máximo.

Questão 5. Considere uma região atravessada por um rio, e suponha que as cidades A_1, A_2, \dots, A_n estão em uma margem do rio, enquanto as cidades B_1, B_2, \dots, B_n se encontram na outra margem do rio. Os moradores desta região desejam construir pontes conectando as cidades e satisfazendo as condições de que duas pontes não podem se cruzar e deve existir pelo menos uma ponte conectando cada cidade ao outro lado do rio. Para cada par de cidades A_i, B_j , o custo de construção de uma ponte entre as duas cidades é denotado por c_{ij} . Deseja-se que o custo total de construção das pontes seja o menor possível.

- (a) Estime o número de soluções válidas para este problema, e verifique que a alternativa de inspecionar todas elas e selecionar a melhor não é viável.
- (b) Proponha um algoritmo guloso para este problema e verifique se ele encontra uma solução ótima.
- (c) Faça um algoritmo de programação dinâmica para encontrar uma solução ótima para o problema. Determine a complexidade do seu algoritmo.