CANA
Lista 03

Victor Campos

1. Uma *subsequência contígua* de uma lista S é uma subsequência de elementos consecutivos de S. Por exemplo, se S é

   $$5, 15, -30, 10, -5, 40, 10$$

   então $15, -30, 10$ é uma subsequência contígua mas $5, 15, 40$ não é. Escreva um algoritmo linear para o seguinte problema:

   Entrada: Uma lista de números $a_1, a_2, \ldots, a_n$.

   Saída: Uma subsequência contígua de soma máxima.

2. Você recebe uma string de n caracteres $s[1, \ldots, n]$ que você acredita ser um texto corrompido em que toda pontuação e espaçamento sumiu ("umtextoescritoassim" ao invés de "um texto escrito assim."). Você gostaria de reconstruir o texto usando um dicionário dicio(w) que recebe como entrada uma string w e retorna um valor booleano indicando se w é uma palavra ou não.

   (a) Escreva um algoritmo de complexidade $\mathrm{O}(n^2)$ que determina se a string s pode ser reconstruída como uma sequência de palavras válidas assumindo que dicio(w) tem complexidade $\mathrm{O}(1)$.

   (b) No evento da string ser válida, faça seu algoritmo retornar a sequência de palavras.

3. Temos um tabuleiro em formato de grid com 4 linhas e n colunas com valores inteiros em cada casa. Gostaríamos de marcar um conjunto de casas deste tabuleiro com soma máxima com a restrição de que não podemos marcar duas casas que são horizontalmente ou verticalmente adjacentes (adjacência diagonal é permitido. Escreva um algoritmo com complexidade $\mathrm{O}(n)$ que calcula uma marcação ótima.

4. Uma subsequência é *palindrômica* se ela é a mesma escrita da esquerda para a direita ou da direita para a esquerda. Por exemplo, a sequência

   $$A, C, G, T, G, T, C, A, A, A, A, T, C, G$$

   tem muitas subsequências palindrômicas, incluindo A, C, G, C, A e A, A, A, A. Escreva um algoritmo com complexidade $\mathrm{O}(n^2)$ que recebe como entrada uma sequência $x[1, \ldots, n]$ e retorna o tamanho da maior subsequência polindrômica.

5. Dados duas strings $x[1, \ldots, n]$ e $y[1, \ldots, m]$, desejamos achar o tamanho da maior subsequência comum entre x e y, isto é, o maior k tal que existem índices $i_1 < i_2 < \cdots < i_k$ e $j_1 < j_2 < \cdots < j_k$ com $x[i_1], x[i_2], \ldots, x[i_k] = y[j_1], y[j_2], \ldots, y[j_k]$. Mostre como fazer isso com complexidade $\mathrm{O}(mn)$.

6. Você está fazendo uma longa viagem. Você começa na estrada no marco zero (quilômetro 0). Ao longo do caminho, há n hotéis localizados nos marcos $a_1 < a_2 < \cdots < a_n$, onde cada a_i é medido a partir do ponto de partida. Os únicos locais em que é permitido parar são esses hotéis, mas você pode escolher em quais deles parar. Você deve obrigatoriamente parar no último hotel (na posição a_n), que é o seu destino.

   Idealmente, você gostaria de viajar 200 quilômetros por dia, mas isso pode não ser possível (dependendo do espaçamento entre os hotéis). Se você viajar x quilômetros em um dia, a *penalidade* para esse dia é $(200 - x)^2$. O objetivo é planejar a viagem de forma a minimizar a penalidade total, ou seja, a soma das penalidades diárias durante toda a viagem.

   Dê um algoritmo eficiente que determine a sequência ótima de hotéis nos quais parar.