

Lista de exercícios 2 (Programação Dinâmica)

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

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

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. Escreva agora um algoritmo de programação dinâmica para o mesmo problema e calcule a complexidade. Escreva também um algoritmo de memoização.

2. Uma subsequência é palíndroma se ela é igual lendo da direita para esquerda ou lendo da esquerda para 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 $S[1 \dots n]$ e retorna a subsequência palíndroma de tamanho máximo.

3. 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, 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 que determina a sequência ótima de hotéis em que você deve parar.

4. 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 de programação dinâmica 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.

5. 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 (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 $j \in \{1, 2, \dots, n\}$, considere subsequências contíguas terminando exatamente na posição j).