

Projeto e Análise de Algoritmos (2025-1)

Flávio L. C. de Moura*

14 de julho de 2025

Exercícios para a prova 2 - gabarito

1. Mostre como podemos ordenar n inteiros contidos no intervalo de 0 a $n^2 - 1$ em tempo linear, ou seja, em $O(n)$.

Solução.

Inicialmente iteramos pela lista dos números, convertendo cada um deles para a base n . Depois aplicamos o algoritmo radix-sort sobre a lista, utilizando o counting-sort como o algoritmo de ordenação dos dígitos da lista de números. Dada a nova base, cada número possuirá 2 dígitos, uma vez que $\log_n(n^2) = 2$, onde cada dígito estará em um intervalo entre 0 e $n - 1$. Sabendo disso, vemos que radix-sort ordenará n números de 2 dígitos, usando o counting-sort em um intervalo de 0 à $n - 1$, resultando em uma complexidade de $O(2(n + n)) = O(n)$.

2. Considere o seguinte problema: Há uma fila de n moedas cujos valores são alguns inteiros positivos $c_1, c_2, \dots, c_n$, não necessariamente distintos. O objetivo é pegar a quantidade máxima de dinheiro sujeita à restrição de que não se pode pegar duas moedas adjacentes na fila inicial.

- (a) Construa uma recorrência para calcular o montante máximo $F(n)$ que pode ser obtido de uma fila com n moedas.

Solução.

$$F(n) = \begin{cases} 0, & \text{se } n = 0 \\ c_1, & \text{se } n = 1 \\ \max(c_n + F(n - 2), F(n - 1)), & \text{se } n > 1 \end{cases}$$

- (b) Construa uma solução (pseudocódigo) utilizando programação dinâmica e faça a análise assintótica da sua solução.

Solução.

A ideia é ir armazenando as soluções dos problemas menores no vetor F (abordagem

*flaviomoura@unb.br

bottom up) de forma que o mesmo subproblema não será computado mais de uma vez. A entrada do algoritmo é o vetor C contendo os valores $c_1, c_2, \dots, c_n$ das moedas.

```

1  $F[0] \leftarrow 0;$ 
2  $F[1] \leftarrow C[1];$ 
3 for  $i = 2$  to  $n$  do
4   |  $F[i] \leftarrow \max(C[i] + F[i - 2], F[i - 1]);$ 
5 end
6 return  $F[n];$ 

```

O algoritmo é linear, isto é, $\Theta(n)$.

3. O problema da mochila possui duas versões:

- (a) O problema da mochila inteira: Dado um conjunto de n objetos com peso w_i e valor v_i $1 \leq i \leq n$, e uma mochila com capacidade de carregar o peso W , onde W, w_i e v_i são inteiros para $1 \leq i \leq n$, quais objetos devem ser colocados na mochila para que o valor total seja máximo?
- (b) O problema da mochila fracionária: Considere n objetos com peso w_i e valor v_i $1 \leq i \leq n$, e uma mochila com capacidade de peso W , de forma que frações de cada objeto podem ser selecionadas. Que fração de cada objeto deve ser colocada na mochila de modo a maximizar o valor total? Em outras palavras, selecione frações $f_i \in [0, 1]$ dos itens tais que $\sum_{i=1}^n f_i \cdot w_i \leq W$ e $\sum_{i=1}^n f_i \cdot v_i$ é máximo.

Podemos utilizar a estratégia gulosa (resp. programação dinâmica) para resolver as duas versões deste problema? Justifique sua resposta.

- 4. Construa um algoritmo eficiente para calcular o coeficiente binomial $C(n, k)$, também denotado por $\binom{n}{k}$, sem utilizar multiplicações. Em seguida, faça a análise assintótica do seu algoritmo.

Solução.

Algorithm 1: Binomial(n, k)

```
1 for  $i = 0$  to  $n$  do
2   for  $j = 0$  to  $\min(i, k)$  do
3     if  $j = 0$  or  $j = i$  then
4        $C[i, j] \leftarrow 1$ ;
5     end
6     else
7        $C[i, j] \leftarrow C[i - 1, j - 1] + C[i - 1, j]$ ;
8     end
9   end
10 end
11 return  $C[n, k]$ ;
```

$$T(n) = \sum_{i=0}^n \left(\sum_{j=0}^{\max(i, k)} 1 \right) \leq \sum_{i=0}^n \left(\sum_{j=0}^k 1 \right) = \sum_{i=0}^n (k+1) = (n+1)(k+1) = O(n.k).$$

5. Um *conjunto independente* em um grafo $G = (V, E)$ é um subconjunto $V' \subseteq V$ de vértices dois a dois não adjacentes. O problema do conjunto independente (INDEPENDENT-SET) consiste em, dados um grafo G e um inteiro k determinar se G possui um conjunto independente de tamanho k . Mostre que INDEPENDENT-SET é NP-difícil. Para isto, transforme uma instância $\varphi = (l_1^1 \vee l_2^1 \vee l_3^1) \wedge (l_1^2 \vee l_2^2 \vee l_3^2) \wedge \dots \wedge (l_1^m \vee l_2^m \vee l_3^m)$ em um grafo G_φ onde cada literal de φ corresponde a um vértice de G_φ , os três vértices que formam uma cláusula de φ são representados por um 3-clique em G_φ , e vértices originados de cláusulas distintas com nomes contraditórios, por exemplo x e $\bar{x}$ são ligados por uma aresta, e conclua que φ é satisfatível se, e somente se, G_φ possui um conjunto independente de tamanho m .