CPS740 - Lista 1

Pedro Maciel Xavier
116023847

July 1, 2020

Questão 1

i) Primeiro Algoritmo

a) Algoritmo

```
seja L[n]

def ordena(L[], n):
    seja m, k
    para i = n até 2:
        m ← −∞
        para j = i até 1:
            se L[j] > m:
                m ← L[j]
                k ← j
        L[j]⟷L[k]
    retorna L
```

b) Passo-a-passo

$$
\begin{aligned}
L = &\{2,7,5,6,\mathbf{9},0,1,4,8,5,\mathbf{3}\}\; 1 \times 11 = 11\\
&\{2,7,5,6,3,0,1,4,\mathbf{8},\mathbf{5},9\}\; 2 \times 10 = 20\\
&\{2,\mathbf{7},5,6,3,0,1,4,\mathbf{5},8,9\}\; 3 \times 9 = 27\\
&\{2,5,5,\mathbf{6},3,0,1,\mathbf{4},7,8,9\}\; 4 \times 8 = 32\\
&\{2,5,\mathbf{5},4,3,0,\mathbf{1},6,7,8,9\}\; 5 \times 7 = 35\\
&\{2,\mathbf{5},1,4,3,\mathbf{0},5,6,7,8,9\}\; 6 \times 6 = 36\\
&\{2,0,1,\mathbf{4},\mathbf{3},5,5,6,7,8,9\}\; 7 \times 5 = 35\\
&\{2,0,1,\mathbf{3},4,5,5,6,7,8,9\}\; 8 \times 4 = 32\\
&\{\mathbf{2},0,\mathbf{1},3,4,5,5,6,7,8,9\}\; 9 \times 3 = 27\\
&\{\mathbf{1},\mathbf{0},2,3,4,5,5,6,7,8,9\}\; 10 \times 2 = 20\\
&\{0,1,2,3,4,5,5,6,7,8,9\}\; \text{Total de 275 passos}
\end{aligned}
$$

ii) Segundo Algoritmo

a) Algoritmo

```
seja L[n]

def ordena(L[], n):
```

Questão 2

Seja $G = (V, E)$ um grafo com m arestas e n vértices.

a) A **Matriz de Adjacências** $A^{n \times n}$ de G é dada por:

$$A_{i,j} = \begin{cases} 1, & \text{se } (i,j) \in E \\ 0, & \text{caso contrário} \end{cases}$$

Desta maneira a matriz terá suas n^2 entradas. Por isso, precisaremos de, no mínimo, n^2 *bits* para representá-lo assim.

b) A **Estrutura de Adjacências** L de G consiste em uma lista de tamanho n contendo em cada nó i um ponteiro para uma lista que contém os vértices adjacentes a i. Ou simplesmente:

$$j \in L_i \iff (i,j) \in E$$

Neste caso, a demanda por espaço não está diretamente relacionada ao número de vértices, mas sim, à quantidade de arestas. Grafos com poucas conexões podem ser representados de maneira eficiente com essa estrutura. Por outro lado, redes muito conexas tornam esta abordagem indesejável.

Para cada aresta introduzida no grafo temos um custo de $2e$ *bits*, onde e representa o armazenamento de um nó j em uma lista e depende da implementação. O custo dobrado vem da necessidade de armazenar cada aresta em duas listas distintas. Por fim, o custo total para armazenar o grafo G desta maneira seria de $2e \times m + v \times n$, considerando v o custo para armazenar L.

c) Comparando as duas escolhas de representação, podemos optar pela **Matriz de Adjacências** sempre que

$$n^2 < 2e \times m + v \times n$$

Questão 3

Questão 4

Questão 5

Questão 6

Questão 7

a) A soma dos quadrados dos números de 1 até n.
b) São executadas sucessivas multiplicações seguidas de soma.
c) Temos n multiplicações e n somas, totalizando $2n$ operações, ou seja, tempo linear $O(n)$.
d) Existem algoritmos melhores. Demonstração:
Temos que

$$\begin{aligned} x^3 - (x-1)^3 &= x^3 - (x^3 - 3x^2 + 3x - 1) \\ &= x^3 - x^3 + 3x^2 - 3x + 1 \\ &= 3x^2 - 3x + 1 \end{aligned}$$

Assim,

$$\begin{aligned} \Longrightarrow \quad & (x-1)^3 - (x-2)^3 = 3(x-1)^2 - 3(x-1) + 1 \\ \Longrightarrow \quad & [x^3 - (x-1)^3] + [(x-1)^3 - (x-2)^3] = [3x^2 - 3x + 1] + [3(x-1)^2 - 3(x-1) + 1] \\ \Longrightarrow \quad & x^3 - (x-2)^3 = 3[x^2 + (x-1)^2] - 3[x + (x-1)] + 2 \\ & \vdots \\ \Longrightarrow \quad & x^3 - (x-(n+1))^3 = 3\sum_{i=1}^{n}(x-i)^2 - 3\sum_{i=1}^{n}(x-i) + (n+1) \end{aligned}$$

Escolhendo $x = 0$:

$$\begin{aligned} \Longrightarrow \quad & -(-(n+1))^3 = 3\sum_{i=1}^{n}(-i)^2 - 3\sum_{i=1}^{n}(-i) + (n+1) \\ \Longrightarrow \quad & (n+1)^3 = 3\sum_{i=1}^{n} i^2 + 3\sum_{i=1}^{n} i + (n+1) \\ \Longrightarrow \quad & \sum_{i=1}^{n} i^2 = \frac{(n+1)^3 - (n+1)}{3} - \sum_{i=1}^{n} i \\ & \qquad = \frac{(n+1)^3 - (n+1)}{3} - \frac{n(n+1)}{2} \end{aligned}$$

Desta maneira, temos 3 somas, 3 multiplicações e 2 divisões, totalizando 8 operações, ou seja, tempo constante $O(1)$.