Projeto e Análise de Algoritmos (2022-1)

Segunda Avaliação

Prof. Flávio L. C. de Moura

20 de setembro de 2022

1. (2.5 pontos) Mostre como podemos ordenar n inteiros contidos no intervalo de 0 a $n^3 - 1$ em tempo linear, ou seja, em tempo $O(n)$.

   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 3 dígitos, uma vez que $\log_n(n^3) = 3$, onde cada dígito estará em um intervalo entre 0 e $n-1$. Sabendo disso, vemos que radix-sort ordenará n números de 3 dígitos, usando o counting-sort em um intervalo de 0 à $n-1$, resultando em uma complexidade de $O(3(n+n)) = O(n)$.

2. (2.5 pontos) Considere o pseudocódigo a seguir:

```
 1 let C[0..h − l] be a new array;
 2 for i = 0 to h − l do
 3 |  C[i] ← 0;
 4 end
 5 for i = 0 to n − 1 do
 6 |  C[A[i] − l] ← C[A[i] − l] + 1;
 7 end
 8 for j = 1 to h − l do
 9 |  C[j] ← C[j] + C[j − 1];
10 end
11 for i = n − 1 downto 0 do
12 |  j ← A[i] − l;
13 |  B[C[j] − 1] ← A[i];
14 |  C[j] ← C[j] − 1;
15 end
16 return B;
```

   Algorithm 1: counting-sort$(A[0..n-1], l, h)$

   Faça a análise da complexidade deste algoritmo.

   Solução:

   Denote $k = h - l$, ou seja, k denota o tamanho do intervalo que contém os elementos a serem ordenado. Assim, o laço **for** das linhas 2-4 é executado em tempo $\Theta(k)$, o laço das linhas 5-7 em tempo $\Theta(n)$, o laço das linhas 8-10 em tempo $\Theta(k)$, e por fim o laço das linhas 11-14 em tempo $\Theta(n)$, o que perfaz um total de $\Theta(n+k)$. Assumindo que $k = O(n)$, temos que o tempo de execução de counting-sort é $\Theta(n)$.

3. Mostre que 3-SAT $\leq_p$ CLIQUE, isto é, mostre que 3-SAT é redutível polinomialmente a CLIQUE considerando a função que transforma instâncias de 3-SAT em instâncias de CLIQUE da seguinte forma:

   Seja φ uma fórmula contendo k cláusulas da seguinte forma:

$$\varphi = (a_1 \vee b_1 \vee c_1) \wedge (a_2 \vee b_2 \vee c_2) \wedge \ldots \wedge (a_k \vee b_k \vee c_k)$$

A função de redução recebe φ como argumento e constrói o par $\langle G, k \rangle$, onde G é um grafo com a seguinte estrutura: os vértice de G são organizados em k grupos de três vértices cada, digamos $t_1, t_2, \ldots, t_k$. Cada tripla t_i $(1 \leq i \leq k)$ corresponde a uma cláusula de φ, e cada vértice da tripla corresponde a um literal da cláusula associada. Marcamos cada vértice com o nome do literal correspondente. Cada aresta de G conecta todos os vértices de G, excetuando-se os seguintes casos:

- vértices que pertencem à mesma tripla;
- vértices cujos nomes são contraditórios, como x e $\overline{x}$

A prova de que a função acima é, de fato, uma redução polinomial é dividida em duas partes:

(a) (2.5 pontos) Mostre que se φ é satisfatível então o grafo G possui um k-clique.

Solução: Suponha que φ seja satisfatível, isto é, existe um designação de variáveis que faz com que todas as k cláusulas sejam verdadeiras. Em cada tripla de G, selecione o vértice correspondente ao literal verdadeiro da cláusula. Se mais de um literal for verdadeiro, então pode-se escolher qualquer um deles. Os vértices de G assim selecionados formam um k-clique. De fato, o número de vértices selecionados é igual a k (um em cada tripla), e cada par de vértices selecionados está ligado por uma aresta já que eles não pertencem à mesma tripla e não possuem nomes contraditórios já que ambos são verdadeiros na designação dada. Logo G possui um k-clique.

(b) (2.5 pontos) Mostre que se G possui um k-clique então φ é satisfatível.

Solução: Suponha que G possui um k-clique. Como cada vértice do clique pertence a uma tripla diferente, construiremos uma designação de variáveis fazendo com que o literal correspondente a cada vértice do clique seja verdadeiro. Observe que é possível construir tal designação porque dois literais contraditórios não correspondem a vértices distintos do clique. Por fim, esta designação satisfaz φ porque cada tripla contém um vértice do clique, e portanto cada cláusula de φ contém um literal verdadeiro.