Laboratório 1 - Blueprints

Erik Perillo, RA135582

18 de setembro de 2016

1 Abordagem

Dados grafo B (do arquivo morto) e A (nome borrado), Duas condições devem ser satisfeitas para serem da mesma cidade:

1. Todos os vértices de B devem estar em A. Basta fazer uma busca exaustiva como descrita no pseudo-código abaixo:

Algorithm 1

1: **procedure** CONTEM(A, B)
2: $\{V_A, E_A\} \leftarrow A, \{V_B, E_B\} \leftarrow B$
3: **if** $|V_B| > |V_A|$ **then return** false
4: **for** v in V_B **do**
5: **if** v $\notin V_A$ **then return** false
return true

2. Para cada vértice (u, v) em B, existe um passeio em A $P = \{u, w_1, \ldots, w_n, v\}$ tal que $\{w_1, \ldots, w_n\}$ não estão em B. A ideia é checar, para toda aresta (u, v) em B, se há um passeio que satisfaz tais condições. `DFS-VISIT` [**clrs**] em A a partir de u pode ser usado para produzir uma lista π de pais. Checa-se então essa lista, a partir de v, indo até u (se existe passeio), verificando se os vértices do meio não estão em B.

Algorithm 2

1: **procedure** NOVOS-VERTICES-EM-VELHAS-CONEXOES(A, B)
2: $\{V_A, E_A\} \leftarrow A, \{V_B, E_B\} \leftarrow B$
3: **for** (u, v) in E_B **do**
4: π = DFS-VISIT(A, u)
5: $w = \pi[v]$
6: **if** $w = NULL$ **then return** false
7: **while** $w \neq u$ **do**
8: **if** $u \in B$ **then return** false
9: $w = \pi[w]$
return true

2 Complexidade

Vamos definir n_1, m_1 o número de vértices e arestas de B, respectivamente, e n_2, m_2 os mesmos de A.

2.1 Algoritmo 1

O primeiro algoritmo na linha 4 executa $O(n_1)$ vezes a linha 5, que por sua vez pode ser implementada em $O(n_2)$. Nas outras linhas, tem tempo constante. Assim, sua complexidade assintótica é $O(n_2 n_1)$ e, como $n_2 \geq n_1$, temos o resultado final de $O(n_2^2)$.

2.2 Algoritmo 2

`DFS-VISIT` na linha 4 executa em $O(m_2)$. O loop da linha 7 é executado em $O(m_2)$ também e a linha 8 pode ser implementada em $O(n_1)$. Todos os passos descritos acima acontecem dentro do *loop* da linha 3 que executa em $O(m_1)$. As outras linhas são em $O(1)$. Assim, o tempo de execução é $O(m_1 (m_2 + m_2 n_1)) = O(m_1 m_2 + m_1 m_2 n_1)$ Sabe-se que há limites entre o número de nós e arestas. Por exemplo:

$$m \leq \frac{n^2 - n}{2}, n \leq 2m$$

Sendo m e n o número de arestas e vértices de um grafo simples qualquer. Assim, $m_1 m_2 + m_1 m_2 n_1 \leq m_1 n_2^2 + m_1 \frac{n_2}{2} n_1 \leq m_1 n_2^2 + m_1 n_2^2$