MO417 – Ata do Exercício 21.3-3

Alisson Soares Pontes

5 de junho de 2010

Enunciado: Dê uma sequência de m operações MAKE-SET, UNION e FIND-SET, n das quais são MAKE-SET, que rode em tempo $\Omega(m \lg n)$ quando usarmos apenas a heurística Union by Rank.

A sequência de operações pode ser a seguinte:

Algorithm 1 EXIGE-M-LG-N(n, k)

for $i \leftarrow 1$ **to** n **do**
    MAKE-SET(x_i)
end for
step $\leftarrow 1$
while *step* $\leq n - 1$ **do**
    for $i \leftarrow 1$ **to** $n -$ *step* **by** $2 *$ *step* **do**
        UNION(x_i, x_{i+step})
    end for
    step $\leftarrow 2 *$ *step*
end while
for $i \leftarrow 1$ **to** k **do**
    FIND-SET($x_{\lfloor \lg n \rfloor}$)
end for

Esta rotina faz n MAKE-SETs, $n - 1$ UNIONs e k FIND-SETs, sendo o total de operações

$$m = n + n - 1 + k,$$

donde

$$k = m - 2n + 1. \qquad (1)$$

Os custos dessas operações estão descritos na Tabela 1, totalizando

$$nc_1 + (n - 1)c_2 + kc_3 \lg n \qquad (2)$$

Número	Operação	custo/operação	custo
n	MAKE-SET	c_1	nc_1
$n-1$	UNION	c_2	$(n-1)c_2$
k	FIND-SET	$c_3 \lg n$	$kc_3 \lg n$

Tabela 1: Número de operações com respectivos pesos individuais e totais. Note que as operações de UNION são realizadas com raízes de árvores, e as operações de FIND-SET com uma folha de profundidade $\lg n$.

operações.

Substituindo (1) em (2) para obter o custo em função de m e n ($f(m,n)$) temos:

$$f(m,n) = c_1 n + c_2 n - c_2 + c_3 m \lg n - 2c_3 n \lg n + c_3 \lg n \tag{3}$$

Como, para uma dada função $g(m,n)$, denota-se $\Omega(g(m,n))$ o conjunto de funções

$$\begin{aligned}\Omega(g(m,n)) = \{h(m,n) : & \text{ existem constantes positivas } c,\ n_0 \text{ e } m_0 \\ & \text{tal que } 0 \leq cg(m,n) \leq h(m,n) \\ & \text{para todo } n \geq n_0 \text{ e } m \geq m_0\end{aligned}$$

Então $f(m,n) = \Omega(m \lg n)$ se existirem constantes positivas c, m_0 e n_0 tais que

$$0 \leq cm \lg n \leq c_1 n + c_2 n - c_2 + c_3 m \lg n - 2c_3 n \lg n + c_3 \lg n, \tag{4}$$

para todo $n \geq n_0$ e $m \geq m_0$

Porém, isto não será possível, já que para $m < n$ nem teremos os n MAKE-SETs exigidos, e para $n \leq m \leq 2n$ não temos certeza se as operações acima darão o tempo mínimo requerido.

Em todo caso, é possível garantir a desigualdade se $m > 3n$, como veremos a seguir.

A desigualdade $0 \leq cm \lg n$ é verdadeira para $m > 0$, $n > 1$ e $c > 0$. Tomando $c = c_3/3$, temos que garantir que

$$\frac{c_3}{3} m \lg n \leq c_1 n + c_2 n - c_2 + c_3 m \lg n - 2c_3 n \lg n + c_3 \lg n,$$

o que equivale a

$$0 \leq c_1 n + c_2 n - c_2 + \frac{2c_3}{3} m \lg n - 2c_3 n \lg n + c_3 \lg n.$$

Mas quando $m > 3n$ o segundo membro da desiguladade acima acaba ficando só com termos positivos ou nulos, desde que $n > 1$.

Então, a sequência m de operações descrita no Algoritmo 1 satisfaz ao enunciado quando $m > 3n$ e $n > 1$.