Análise de Algoritmos

Parte destes slides são adaptações de slides

do Prof. Paulo Feofiloff e do Prof. José Coelho de Pina.

Análise do Union-Find

CLRS cap 21

Coleção de conjuntos disjuntos

Queremos uma ED boa para representar uma partição de um conjunto, e as seguintes operações sobre a partição:

- MAKESET(x): cria um conjunto unitário com o elemento x;
- FINDSET(x): devolve o identificador do conjunto da partição que contém x;
- UNION(x, y): substitui os conjuntos da partição que contêm x e y pela união deles.

O identificador de um conjunto é um elemento do conjunto: o seu representante.

Como podemos armazenar cada conjunto da partição?

Implementação 1 do union-find

Make-Set (x)
    pai$[x] \leftarrow x$

Find (x)
    $r \leftarrow x$
    enquanto pai$[r] \neq r$ **faça**
        $r \leftarrow$ pai$[r]$
    devolva r

Union (x, y) $\triangleright$ x e y representantes distintos
    pai$[y] \leftarrow x$

Consumo de tempo: do Find pode ser muito ruim... $\Theta(n)$.
Temos que fazer melhor...

Implementação 2

Heurística dos tamanhos

Make-Set (x)
```
pai[x] ← x
rank[x] ← 0
```

Find (x): o mesmo de antes

Union (x, y) ▷ x e y representantes distintos
```
se rank[x] ≥ rank[y]
    então pai[y] ← x
          se rank[x] = rank[y]
             então rank[x] ← rank[x] + 1
    senão pai[x] ← y
```

Consumo de tempo: melhor... $\Theta(\lg n)$.
Dá para fazer melhor ainda!

Implementação 3

Heurística da compressão dos caminhos

Find (x)
if pai$[x] \neq x$
  então pai$[x] \leftarrow$ Find (pai$[x]$)
devolva pai$[x]$

Consumo *amortizado* de tempo de cada operação:

$$\mathrm{O}(\lg^* n),$$

onde $\lg^* n$ é o número de vezes que temos que aplicar o $\lg$ até atingir um número menor ou igual a 1.

Na verdade, é melhor do que isso,
e há uma análise justa, conforme discutido em aula.

Union-Find

Make-Set (x)

pai$[x] \leftarrow x$
rank$[x] \leftarrow 0$

Find (x)

if pai$[x] \neq x$
    então pai$[x] \leftarrow$ Find (pai$[x]$)
devolva pai$[x]$

Union (x, y) $\qquad \triangleright$ x e y representantes distintos

se rank$[x] \geq$ rank$[y]$
    então pai$[y] \leftarrow x$
        se rank$[x] =$ rank$[y]$
            então rank$[x] \leftarrow$ rank$[x] + 1$
    senão pai$[x] \leftarrow y$

Union-Find

Union (x, y)
$x' \leftarrow$ Find (x)
$y' \leftarrow$ Find (y)
se $x' \neq y'$
  então Link (x', y')

Link (x, y) $\triangleright$ x e y representantes distintos
se rank$[x] \geq$ rank$[y]$
  então pai$[y] \leftarrow x$
    se rank$[x] =$ rank$[y]$
      então rank$[x] \leftarrow$ rank$[x] + 1$
  senão pai$[x] \leftarrow y$

Consumo de tempo

Dada sequência de MAKESET, FINDSET e UNION,
converta-a em uma sequência de MAKESET, FINDSET e LINK.

Sequência de m operações MAKESET, FINDSET e LINK
das quais n são MAKESET.

Custo de pior caso de cada operação: $\mathrm{O}(\lg n)$.
Custo amortizado de cada operação: $\mathrm{O}(\lg^* n)$.

Para definir $\lg^* n$, seja $\lg^{(1)} x = \lg x$.
Para $i \geq 2$, seja $\lg^{(i)} x = \lg(\lg^{(i-1)} x)$.

Então $\lg^* n \ = \ \min\{i : \lg^{(i)} n \leq 1\}$.

A análise desta ED é vista na disciplina MAC6711.