MAC 338 - Análise de Algoritmos

Departamento de Ciência da Computação

Primeiro semestre de 2005

Conjuntos disjuntos

Seja U um conjunto com n elementos e $\mathcal{S}$ uma partição de U. Como representar $\mathcal{S}$ de maneira a ter uma implementação o mais eficiente possível das três operações abaixo?

- **MAKESET**(x): dado um elemento x de U, cria um conjunto contendo apenas x;
- **FINDSET**(x): dado um elemento x de U, devolve o conjunto da partição $\mathcal{S}$ ao qual x pertence;
- **UNION**(x, y): dados elementos x e y de U distintos, substitui em $\mathcal{S}$ os conjuntos que contêm x e y pela união destes conjuntos.

Na verdade, a operação **FINDSET**(x) devolve um dos elementos do conjunto ao qual x pertence, que é o *representante* do conjunto.

Para facilitar a análise das operações acima, consideramos a seguinte operação extra:

- **LINK**(x, y): dados elementos x e y de U distintos, cada um representante de um dos conjuntos da partição $\mathcal{S}$, substitui em $\mathcal{S}$ os conjuntos representados por x e y pela sua união.

Com ela, a operação **UNION** pode ser implementada da seguinte maneira.

```
UNION (x, y)
1   se FINDSET(x) ≠ FINDSET(y)
2       então LINK(FINDSET(x),FINDSET(y))
```

Precisamos analisar então apenas o custo das operações **MAKESET**, **FINDSET** e **LINK**.

Idéia 1: Cada conjunto é representado por uma lista ligada circular.

O representante de um conjunto recebe uma marca. Dessa maneira pode-se implementar o **MAKESET**(x), **FINDSET**(x) e o **LINK**(x, y) de modo a consumirem no pior caso, respectivamente, $\Theta(1)$, $\Theta(n)$ e $\Theta(1)$. (Veja o exercício 1.)

Idéia 2: Cada conjunto é representado por uma árvore enraizada de altura 1, ou seja, cada elemento do conjunto aponta para o representante do conjunto, que é a raiz da árvore.

Desta maneira, pode-se implementar o **MAKESET**(x), **FINDSET**(x) e o **LINK**(x, y) de modo a consumirem no pior caso, respectivamente, $\Theta(1)$, $\Theta(1)$ e $\Theta(n)$. (Veja o exercício 2.)

Pode-se fazer um pequeno mas valioso refinamento nessa idéia que baixa não o custo de pior caso do **LINK**(x, y), mas o seu custo amortizado, sem aumentar o custo das outras operações. A idéia é, para cada conjunto, guardar também o tamanho corrente do conjunto (como um campo extra na célula do representante do conjunto, por exemplo). Usando esta informação, pode-se implementar o **LINK**(x, y) mais cuidadosamente, garantindo que o consumo de tempo seja proporcional ao tamanho do menor dos dois conjuntos. Ainda assim, no pior caso o **LINK**(x, y) pode custar $\Theta(n)$, pois podemos estar unindo dois conjuntos com $n/2$ elementos. Mas uma análise amortizada mostra que o custo amortizado do **LINK**(x, y) cai para $O(\lg n)$.

De fato, considere uma seqüência de m operações **MAKESET**, **LINK** e **FINDSET**, onde n delas são **MAKESET** (e portanto não mais que $n-1$ delas são **LINK**). O custo agregado de todas as operações **LINK** na seqüência é proporcional ao número destas operações mais o número total de vezes que um apontador para o representante é alterado durante as operações **LINK**. Se olharmos uma célula específica, o seu apontador para representante pode ser alterado no máximo $\lfloor \lg n \rfloor$ vezes, pois, a cada alteração, seu conjunto pelo menos duplica de tamanho. Sendo assim, são feitas não mais que $n \lg n$ alterações dessas e o custo total das m operações é $O(m + n \lg n)$, que resulta em um custo amortizado de $O(\lg n)$ para o **LINK** e de $O(1)$ para o **MAKESET** e para o **FINDSET**.

Idéia 3: ...

Exercícios

1. Mostre que o custo amortizado das operações **FINDSET** e **LINK**, se implementadas conforme a idéia 1, não é melhor que o custo de pior caso.

2. Mostre que o custo amortizado das operações **FINDSET** e **LINK**, se implementadas conforme a idéia 2 (sem refinamentos), não é melhor que o custo de pior caso.