



Estrutura de Dados I

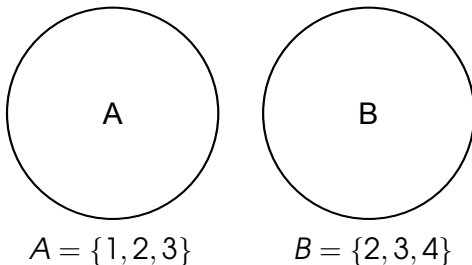
Conjuntos disjuntos

Bruno Prado

Departamento de Computação / UFS

Introdução

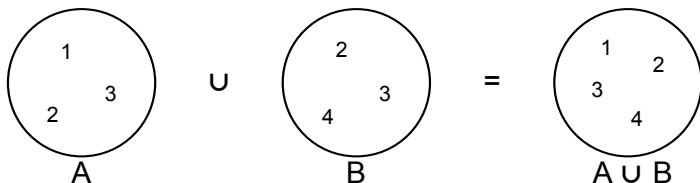
- ▶ O que é um conjunto?
 - ▶ É uma coleção de elementos distintos
 - ▶ Podem ser ilustrados através do diagrama de Venn



Introdução

- ▶ Operação de união de conjuntos

- ▶ Notação \cup



$$A = \{1, 2, 3\}$$

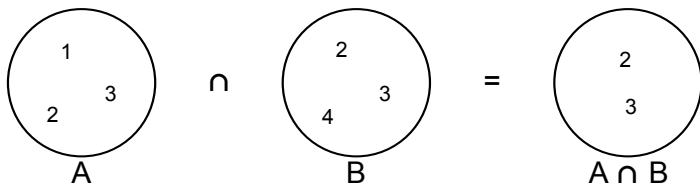
$$B = \{2, 3, 4\}$$

$$A \cup B = \{1, 2, 3, 4\}$$

Introdução

- ▶ Operação de interseção de conjuntos

- ▶ Notação \cap



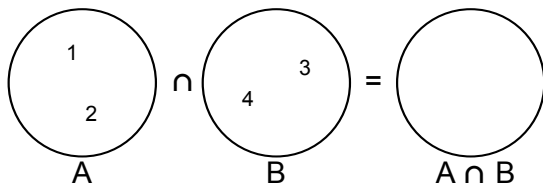
$$A = \{1, 2, 3\}$$

$$B = \{2, 3, 4\}$$

$$A \cap B = \{2, 3\}$$

Introdução

- ▶ O que são conjuntos disjuntos?
 - ▶ São coleções de elementos que não possuem interseção entre si



$$A = \{1, 2\}$$

$$B = \{3, 4\}$$

$$A \cap B = \emptyset$$

Introdução

- ▶ Operações básicas em conjuntos disjuntos
 - ▶ Criação do conjunto (make-set)
 - ▶ União de conjuntos (union)
 - ▶ Busca de conjunto (find-set)

Devido a estas operações principais, esta estrutura de dados também é conhecida como conjunto union-find

Conjuntos Disjuntos

- ▶ Definição da estrutura de conjuntos disjuntos
 - ▶ Coleção de conjuntos disjuntos $S = \{S_1, S_2, \dots, S_n\}$
 - ▶ Cada conjunto é identificado por um representante
 - ▶ Este representante é algum elemento do conjunto
 - ▶ Não importa qual é o elemento
 - ▶ Só precisa ser o mesmo em todas as operações
- ▶ Estruturas de armazenamento
 - ▶ Listas encadeadas
 - ▶ Árvores

Conjuntos Disjuntos

- ▶ Criação de um conjunto disjunto (make set)
 - ▶ É feita a criação de um conjunto i com exatamente um elemento $S_i = \{x\}$
 - ▶ O elemento x é o representante do conjunto, só existindo neste conjunto disjunto
 - ▶ Este novo conjunto criado é adicionado a coleção de conjuntos disjuntos $S = S \cup S_i$

Cada elemento é inserido através desta operação, sendo realizadas n operações para criação dos conjuntos disjuntos

Conjuntos Disjuntos

- ▶ União de dois conjuntos disjuntos (union)
 - ▶ Assumindo dois conjuntos $S_i = \{x\}$ e $S_j = \{y\}$, é necessário escolher um novo representante de S_i ou S_j
 - ▶ É feita a remoção dos conjuntos S_i e S_j dos conjuntos disjuntos $S = S - S_i - S_j$
 - ▶ A operação de união cria um novo conjunto $S_k = S_i \cup S_j$ que é incorporado à coleção $S = S \cup S_k$

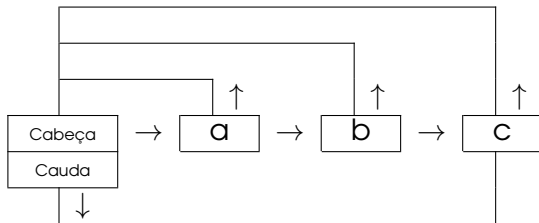
Cada operação de união reduz a quantidade de conjuntos disjuntos por 1, realizando no máximo $n - 1$ uniões até que reste apenas um conjunto

Conjuntos Disjuntos

- ▶ Busca por um conjunto disjunto (find-set)
 - ▶ É retornado o conjunto que contém o elemento x , através da referência do conjunto
 - ▶ Esta referência contém o representante do conjunto que contém o elemento x

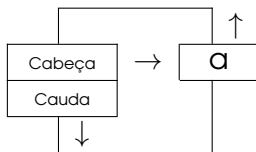
Conjuntos Disjuntos

- Representação por lista encadeada
 - Cada conjunto é uma lista encadeada, com ponteiros para cabeça e para cauda da lista
 - Os elementos apontam para o próximo elemento e para cabeça da lista que é o representante



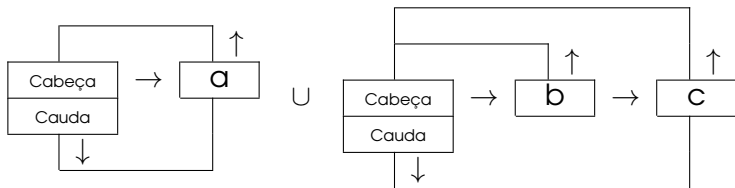
Conjuntos Disjuntos

- ▶ Representação por lista encadeada
 - ▶ Criando um conjunto para o elemento a
 - ▶ A criação da lista possui custo $O(1)$



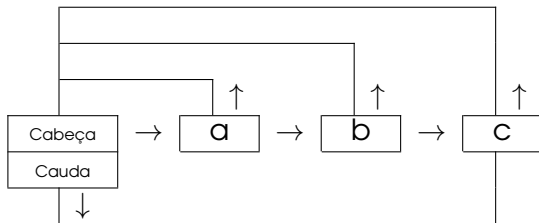
Conjuntos Disjuntos

- Representação por lista encadeada
 - Unindo dois conjuntos disjuntos
 - Para realizar a união dos elementos a , b e c é necessário ajustar os ponteiros



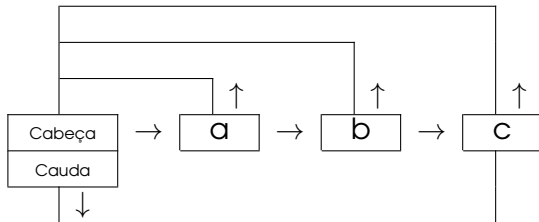
Conjuntos Disjuntos

- Representação por lista encadeada
 - Unindo dois conjuntos disjuntos
 - O ajuste dos ponteiros tem custo $O(n)$



Conjuntos Disjuntos

- Representação por lista encadeada
 - Buscando o conjunto do elemento c
 - O custo desta operação é $O(1)$



Conjuntos Disjuntos

- Representação por lista encadeada
 - Considerando os conjuntos disjuntos $S = \{S_1, S_2, \dots, S_n\}$
 - Realizando n criações e $n - 1$ uniões de conjuntos

| Operação | Número de Atualizações |
|-------------------------|------------------------|
| Criação(S_1) | 1 |
| Criação(S_2) | 1 |
| \vdots | 1 |
| Criação(S_n) | 1 |
| União(S_1, S_2) | 1 |
| União(S_2, S_3) | 2 |
| \vdots | \vdots |
| União(S_{n-1}, S_n) | $n - 1$ |

$$O\left(n + \sum_{i=1}^{n-1} i\right) = O(n + n^2) = O(n^2)$$

Conjuntos Disjuntos

- ▶ Representação por lista encadeada
 - ▶ Para tornar a estrutura mais eficiente, é aplicada uma heurística de união ponderada
 - ▶ É feita a união do conjunto de menor tamanho com o que possui maior tamanho, atualizando no pior caso metade das referências da lista



- ▶ A união de n elementos tem custo de $O(n \log_2 n)$

Conjuntos Disjuntos

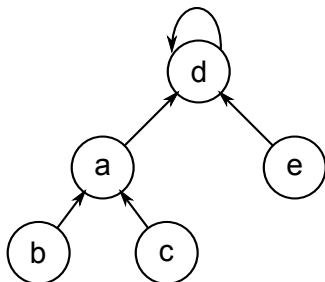
- ▶ Análise de complexidade
 - ▶ Considerando que o custo para realização de m operações constantes para criação e busca de conjuntos disjuntos é $O(m)$
 - ▶ Espaço $\Theta(n)$
 - ▶ Tempo $O(m + n \log_2 n)$

Exemplo

- ▶ Realize a criação de conjuntos para os elementos 1, 2, 3, 4, 5, 6, ilustrando as estruturas de lista a medida que as operações abaixo são executadas
 - ▶ União(1, 2)
 - ▶ União(3, 4)
 - ▶ União(5, 6)
 - ▶ União(1, 6)
 - ▶ Busca(5)
 - ▶ União(3, 5)

Conjuntos Disjuntos

- Representação por árvore
 - Cada conjunto é representado por uma árvore enraizada criando uma floresta de conjuntos disjuntos
 - Na implementação das referências o nó possui uma referência para o elemento pai, referenciando a si mesmo quando for raiz da árvore



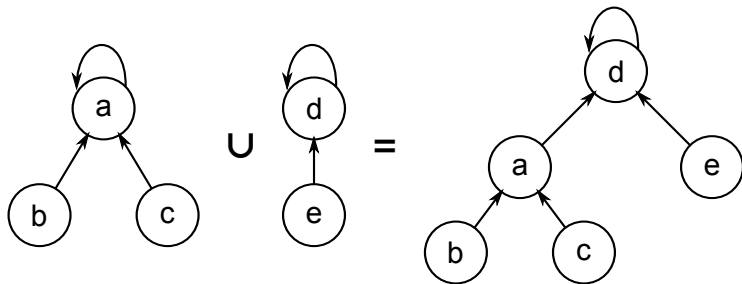
Conjuntos Disjuntos

- ▶ Representação por árvore
 - ▶ Criando um conjunto para o elemento a
 - ▶ A criação da árvore possui custo $O(1)$



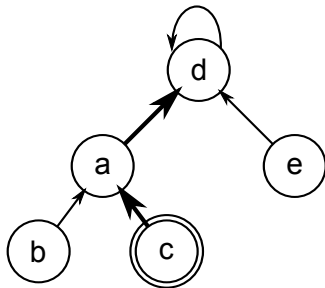
Conjuntos Disjuntos

- Representação por árvore
 - Unindo dois conjuntos disjuntos
 - O ajuste dos ponteiros tem custo $O(1)$



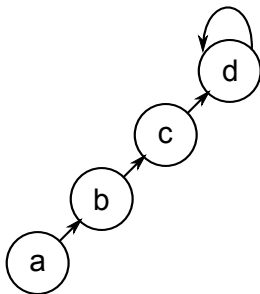
Conjuntos Disjuntos

- Representação por árvore
 - Buscando o conjunto do elemento c
 - O custo desta operação é $O(h)$



Conjuntos Disjuntos

- ▶ Representação por árvore
 - ▶ Considerando os conjuntos disjuntos $S = \{S_1, S_2, \dots, S_n\}$
 - ▶ Realizando n criações e $n - 1$ uniões de conjuntos
 - ▶ No pior caso $O(n^2)$, a árvore é uma lista encadeada

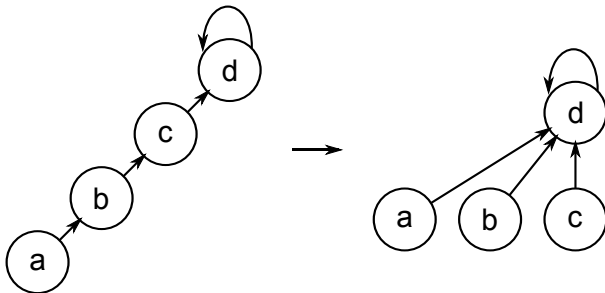


Conjuntos Disjuntos

- ▶ Representação por árvore
 - ▶ Para melhorar a eficiência da estrutura é aplicada a heurística de união por classificação
 - ▶ Nesta heurística de união, cada árvore é classificada pelo número de nós que possui e a árvore com menor número de nós fará referência para a árvore com maior número de nós
 - ▶ Considerando m operações de criação e de união de conjuntos, esta heurística tem custo $O(m \log_2 n)$

Conjuntos Disjuntos

- ▶ Representação por árvore
 - ▶ A heurística de compressão de caminhos permite tornar a estrutura ainda mais eficiente, fazendo que os nós referenciem diretamente a raiz da árvore
 - ▶ Não é alterada a quantidade de nós de cada árvore



Conjuntos Disjuntos

- ▶ Implementação em C
 - ▶ Na criação de um conjunto, o nó referencia a si mesmo e possui altura nula

```
void make-set(node* x) {  
    x->p = x;  
    x->rank = 0;  
}
```

Conjuntos Disjuntos

- Implementação em C

- A união de dois conjuntos é feita pela aplicação da heurística de união por classificação

```
void union(node* x, node* y) {  
    node* rx = find-set(x);  
    node* ry = find-set(y);  
    if(rx->rank > ry->rank)  
        ry->p = rx;  
    else {  
        rx->p = ry;  
        if(rx->rank == ry->rank)  
            ry->rank++;  
    }  
}
```

Conjuntos Disjuntos

- Implementação em C
 - Na busca pelo conjunto é feita a aplicação da heurística de compressão de caminhos

```
node* find-set(node* x) {  
    if(x != x->p)  
        x->p = find-set(x->p);  
    return x->p;  
}
```

Conjuntos Disjuntos

- ▶ Análise de complexidade
 - ▶ Aplicando as duas heurísticas de união por classificação e de compressão de caminhos é obtido um tempo de execução $O(m\alpha(n))$, a função $\alpha(n)$ possui um crescimento muito lento e utilizando valores práticos de n a função $\alpha(n) \leq 4$
 - ▶ Espaço $\Theta(n)$
 - ▶ Tempo $O(m)$

Exemplo

- ▶ Realize a criação de conjuntos para os elementos 1, 2, 3, 4, 5, 6, ilustrando as estruturas de árvore a medida que as operações abaixo são executadas
 - ▶ União(1, 2)
 - ▶ União(3, 4)
 - ▶ União(5, 6)
 - ▶ União(1, 6)
 - ▶ Busca(5)
 - ▶ União(3, 5)

Conjuntos Disjuntos

► Aplicações

- Árvore de extensão mínima (Kruskal)
- Particionamento de conjuntos
- Verificar conectividade dos nós de uma rede
- ...

