Conjuntos Disjuntos

Diego Samir Melo Solarte – IC/UNICAMP

8 de outubro de 2007

Introdução

Introdução

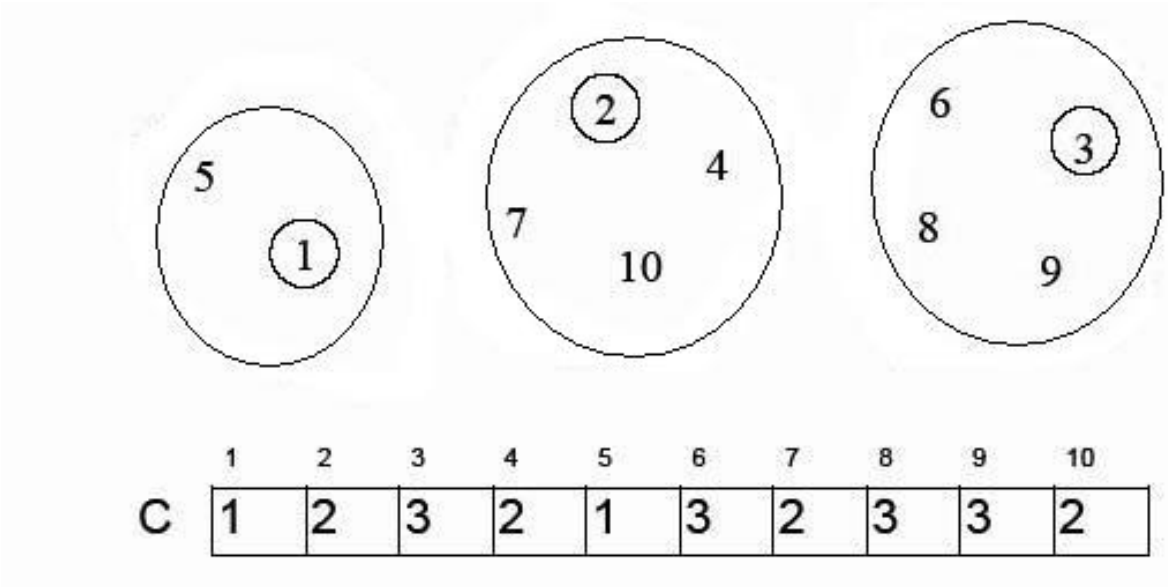
- Uma estrutura de dados conjuntos-disjuntos é uma coleção $S = \{S_1, ..., S_k\}$ de conjuntos dinâmicos disjuntos.

Introdução

- Uma estrutura de dados conjuntos-disjuntos é uma coleção $S = \{S_1, ..., S_k\}$ de conjuntos dinâmicos disjuntos.
- Cada conjunto S_k é identificado por um representante, que é um membro do conjunto.

Introdução

- Uma estrutura de dados conjuntos-disjuntos é uma coleção $S = \{S_1, ..., S_k\}$ de conjuntos dinâmicos disjuntos.
- Cada conjunto S_k é identificado por um representante, que é um membro do conjunto.
- Tipicamente não importa quem é o representante, apenas que ele seja consistente.



Operações

Operações

- Make_Set(x): Cria um novo conjunto cujo único elemento é apontado por x. x não pode pertencer a outro conjunto da coleção.

Operações

- Make_Set(x): Cria um novo conjunto cujo único elemento é apontado por x. x não pode pertencer a outro conjunto da coleção.
- Union(x, y): Executa a união dos conjuntos que contêm x e y, digamos Sx e Sy, em um conjunto único.
  $Sx \bigcap Sy = 0$.
  O representante de $S = Sx \bigcup Sy$ é um elemento de S.

Operações

- Make_Set(x): Cria um novo conjunto cujo único elemento é apontado por x. x não pode pertencer a outro conjunto da coleção.
- Union(x, y): Executa a união dos conjuntos que contêm x e y, digamos Sx e Sy, em um conjunto único.
  $Sx \bigcap Sy = 0$.
  O representante de $S = Sx \bigcup Sy$ é um elemento de S.
- Find(x): Retorna um ponteiro para o representante (único) do conjunto que contém x.

Aplicaçãoes

Aplicaçãoes

- Algumas aplicações envolvem o agrupamento de N elementos em uma coleção de conjuntos disjuntos, ou seja, um particionamento dos elementos em conjuntos.

Aplicaçãoes

- Algumas aplicações envolvem o agrupamento de N elementos em uma coleção de conjuntos disjuntos, ou seja, um particionamento dos elementos em conjuntos.

- Alguns usos:
  - problemas de grafos
  - equivalência de tipos em compiladores

Exemplo de Aplicação: Componentes Conexos

Exemplo de Aplicação: Componentes Conexos

- Uma aplicação da estrutura de dados conjuntos disjuntos surge na determinação dos componentes conexos de um grafo.

Connected_Components(G)

1. for each vertex $v \in V[G]$
2. do Make_Set(v)
3. for each $(u, v) \in E$
4. do if Find_Set(u) $\neq$ Find_Set(v)
5. then Union(u,v)

Exemplo de Aplicação: Componentes Conexos

Exemplo de Aplicação: Componentes Conexos

- Uma aplicação da estrutura de dados conjuntos disjuntos surge na determinação dos componentes conexos de um grafo.

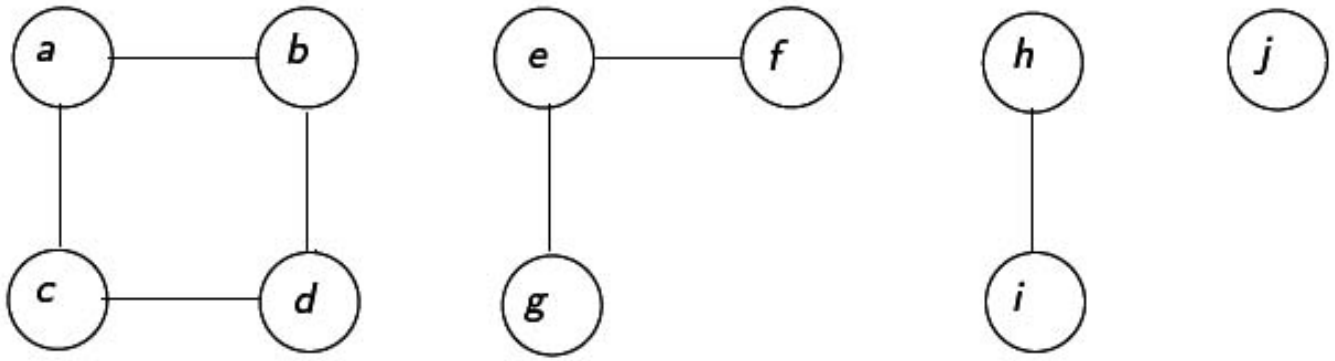
Same_Component(u, v)

1. if Find_Set(u) = Find_Set(v)
2. then return True
3. else return False

Exemplo de Aplicação: Componentes Conexos

Exemplo de Aplicação: Componentes Conexos

- Uma aplicação da estrutura de dados conjuntos disjuntos surge na determinação dos componentes conexos de um grafo.



Coleção de Conjuntos Disjuntos

Início	$\{a\}$	$\{b\}$	$\{c\}$	$\{d\}$	$\{e\}$	$\{f\}$	$\{g\}$	$\{h\}$	$\{i\}$	$\{j\}$
(b, d)	$\{a\}$	$\{b, d\}$	$\{c\}$		$\{e\}$	$\{f\}$	$\{g\}$	$\{h\}$	$\{i\}$	$\{j\}$
(e, g)	$\{a\}$	$\{b, d\}$	$\{c\}$		$\{e, g\}$	$\{f\}$		$\{h\}$	$\{i\}$	$\{j\}$
(a, c)	$\{a, c\}$	$\{b, d\}$			$\{e, g\}$	$\{f\}$		$\{h\}$	$\{i\}$	$\{j\}$
(h, i)	$\{a, c\}$	$\{b, d\}$			$\{e, g\}$	$\{f\}$		$\{h, i\}$		$\{j\}$
(a, b)	$\{a, b, c, d\}$				$\{e, g\}$	$\{f\}$		$\{h, i\}$		$\{j\}$
(e, f)	$\{a, b, c, d\}$				$\{e, f, g\}$			$\{h, i\}$		$\{j\}$
(c, d)	$\{a, b, c, d\}$				$\{e, f, g\}$			$\{h, i\}$		$\{j\}$

Representação de Conjuntos Disjuntos

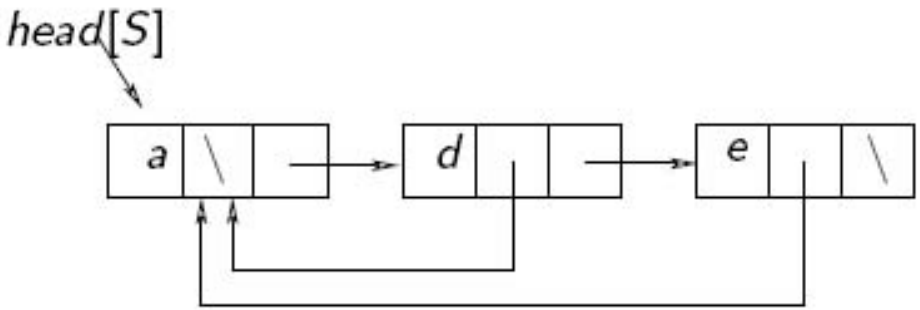
Representação de Conjuntos Disjuntos

- Uma maneira simples de implementar uma estrutura de dados Conjuntos Disjuntos consiste em representar cada conjunto como uma lista encadeada.

Representação de Conjuntos Disjuntos

- Uma maneira simples de implementar uma estrutura de dados Conjuntos Disjuntos consiste em representar cada conjunto como uma lista encadeada.
- O primeiro elemento da lista é o representante.

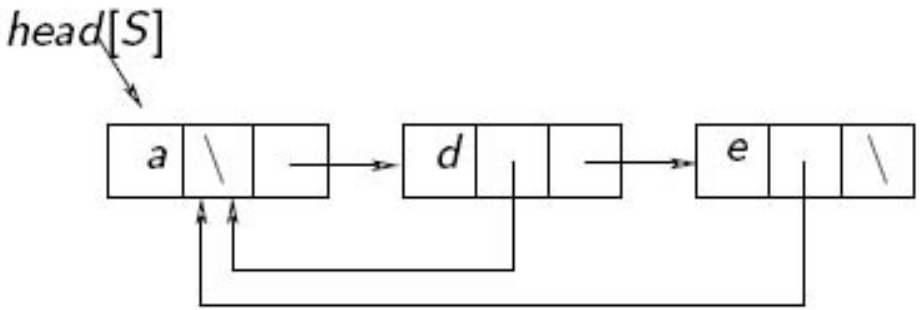
Exemplo: $S = \{a, d, e\}$



Representação de Conjuntos Disjuntos

- Uma maneira simples de implementar uma estrutura de dados Conjuntos Disjuntos consiste em representar cada conjunto como uma lista encadeada.
- O primeiro elemento da lista é o representante.

Exemplo: $S = \{a, d, e\}$

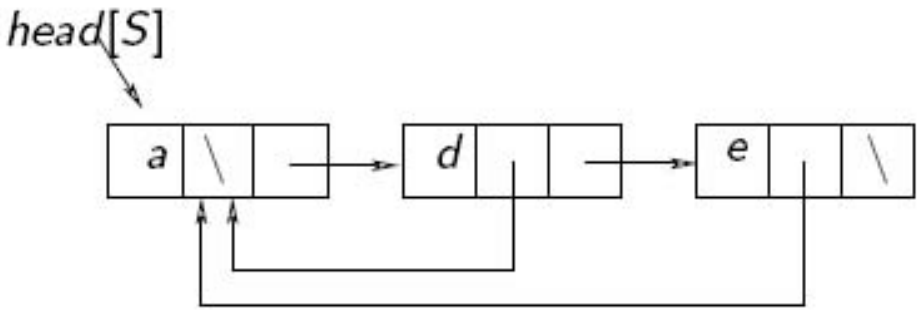


- $Find\_Set \in O(1)$

Representação de Conjuntos Disjuntos

- Uma maneira simples de implementar uma estrutura de dados Conjuntos Disjuntos consiste em representar cada conjunto como uma lista encadeada.
- O primeiro elemento da lista é o representante.

Exemplo: $S = \{a, d, e\}$



- $Find\_Set \in O(1)$
- $Make\_Set \in O(1)$

Implementação de Conjuntos Disjuntos

Conjuntos Disjuntos com Listas Encadeadas

Implementação de Conjuntos Disjuntos

Conjuntos Disjuntos com Listas Encadeadas

- A forma mais simples de implementar a operação união, Union(x, y), é adicionar a lista de x no fim da lista de y.

Implementação de Conjuntos Disjuntos

Conjuntos Disjuntos com Listas Encadeadas

- A forma mais simples de implementar a operação união, Union(x, y), é adicionar a lista de x no fim da lista de y.
- O representante do conjunto é o elemento que era originalmente o representante de y.

Implementação de Conjuntos Disjuntos

Conjuntos Disjuntos com Listas Encadeadas

- A forma mais simples de implementar a operação união, Union(x, y), é adicionar a lista de x no fim da lista de y.
- O representante do conjunto é o elemento que era originalmente o representante de y.
- Todavia, temos de atualizar o ponteiro para o representante de todos os elementos de x.

Implementação de Conjuntos Disjuntos

Conjuntos Disjuntos com Listas Encadeadas

- A forma mais simples de implementar a operação união, Union(x, y), é adicionar a lista de x no fim da lista de y.
- O representante do conjunto é o elemento que era originalmente o representante de y.
- Todavia, temos de atualizar o ponteiro para o representante de todos os elementos de x.
- Não é difícil construir uma instância e seqüência de n operações que exija o tempo $\theta(n^2)$.

Implementação de Conjuntos Disjuntos

instância de exemplo:

Implementação de Conjuntos Disjuntos

instância de exemplo:

- Seja n o número de operações Make_Set

Implementação de Conjuntos Disjuntos

instância de exemplo:

- Seja n o número de operações Make_Set
- Seja m o número total de operações Union, Make_Set e Find_Set

Implementação de Conjuntos Disjuntos

instância de exemplo:

- Seja n o número de operações Make_Set
- Seja m o número total de operações Union, Make_Set e Find_Set
- Suponha que tenhamos $X_1, ..., X_n$ objetos

Implementação de Conjuntos Disjuntos

instância de exemplo:

- Seja n o número de operações Make_Set
- Seja m o número total de operações Union, Make_Set e Find_Set
- Suponha que tenhamos $X_1, ..., X_n$ objetos
- Então executamos uma seqüência de n operações Make_Set seguidas por $n-1$ operações Union, de forma que $m = 2n-1$.

Operação	Número de objetos atualizados
Make_Set(x_1)	1
Make_Set(x_2)	1
⋮	⋮
Make_Set(x_n)	1
Union(x_1, x_2)	1
Union(x_2, x_3)	2
Union(x_3, x_4)	3
⋮	⋮
Union(x_{n-1}, x_n)	$\mathbf{n-1}$

Representação de Conjuntos Disjuntos

Representação de Conjuntos Disjuntos

- Gastamos o tempo de $\theta(n)$ executando as n operações Make_Set

Representação de Conjuntos Disjuntos

- Gastamos o tempo de $\theta(n)$ executando as n operações Make_Set
- Pelo fato da i-ésima operação Union atualizar i objetos, o numero total de objetos atualizados por todas as $n-1$ operações Union é $\theta(n^2)$

Representação de Conjuntos Disjuntos

- Gastamos o tempo de $\theta(n)$ executando as n operações Make_Set
- Pelo fato da i-ésima operação Union atualizar i objetos, o numero total de objetos atualizados por todas as $n-1$ operações Union é $\theta(n^2)$
- O número de operações $\theta(n + \sum_{i=1}^{n-1} i) = \theta(n + n^2)$

Heurística de União Ponderada

Heurística de União Ponderada

- De acordo com a implementação acima, cada operação leva tempo médio $\theta(m)$ por que talvez estejamos inserindo uma lista mais longa em uma lista pequena.

Heurística de União Ponderada

- De acordo com a implementação acima, cada operação leva tempo médio $\theta(m)$ por que talvez estejamos inserindo uma lista mais longa em uma lista pequena.

- Heurística:

  - Cada representante armazena o comprimento da lista de elementos do seu conjunto
  - A inserção é sempre feita da lista menor na lista maior

Heurística de União Ponderada

Teorema 21.1:

Usando a representação de lista encadeada e a heurística de união ponderada, uma seqüência de m operações Make_Set, Union e Find_Set, dentre as quais n são operaçoes Make_Set, executa em tempo O(m + n log n).

Heurística de União Ponderada

Heurística de União Ponderada

Prova:

Vamos calcular para cada objeto, em um conjunto n, um limite superior para o número de vezes que o ponteiro para o representante foi atualizado.

- Toda vez que o ponteiro para o representante x, p[x] foi atualizado, x deveria estar em uma lista menor.
- A primeira vez, o conjunto resultante deveria ter 2 elementos
- A segunda vez, o conjunto resultante deveria ter pelo menor 4 elemento
- Para qualquer $k \leq n$, após p[x] ter sido atualizado $\lfloor logk \rfloor$, o conjunto de x deveria ter pelo menos k elementos

Heurística de União Ponderada

Heurística de União Ponderada

Prova Continuação:

- Uma vez que o conjunto mais longo tem n elementos, cada ponteiro para o representante é atualizado no máximo $\lfloor logn \rfloor$ vezes
- Portanto, o tempo total gasto na atualização de n ponteiros é O(n log n)
- O tempo de execução de uma seqüência completa de m operações pode ser calculada como segue:
  - Make_Set e Find_Set são O(1), existindo no máximo O(m) operações desta natureza
  - Portanto, o tempo total é O(m + n log n)

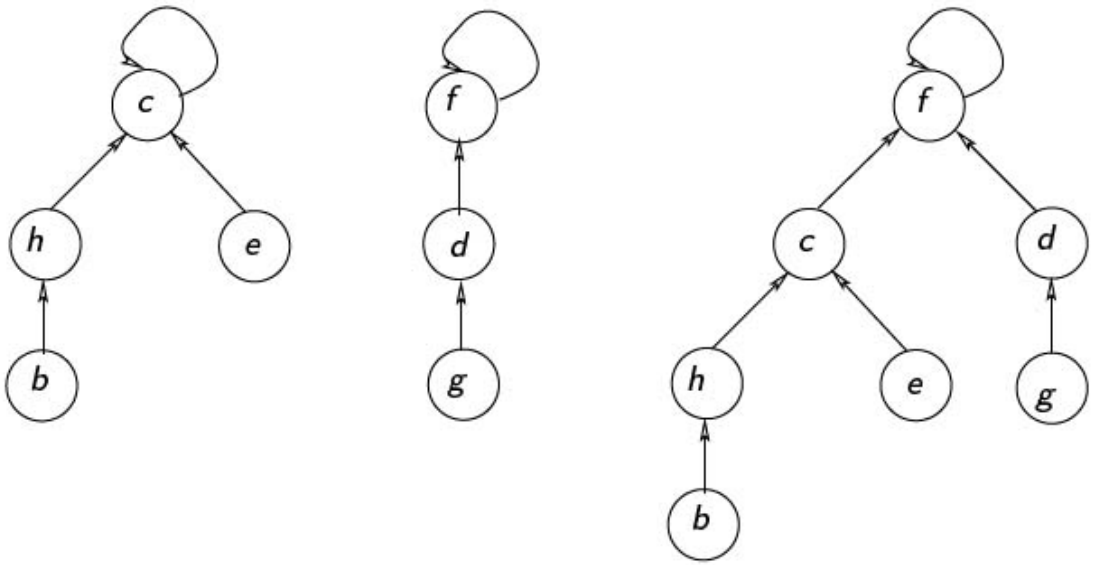
Florestas de Conjuntos Disjuntos

Florestas de Conjuntos Disjuntos

Em uma implementação mais rápida da estrutura de dados Conjuntos Disjuntos, representamos conjuntos por meio de árvores.

- A raiz da árvore contém o representante do conjunto.
- O filho aponta para o pai.
- AGM - Kruskal

$Sc = \{c, h, e, b\}$ e $Sf = \{f, d, g\}$



Florestas de Conjuntos Disjuntos

Florestas de Conjuntos Disjuntos

A raiz da árvore contém o representante do conjunto

Florestas de Conjuntos Disjuntos

A raiz da árvore contém o representante do conjunto

Na forma apresentada, a estrutura é tão lenta quanto aquela que utiliza listas encadeadas.

- Make_Set: cria uma árvore com um único vértice.
- Find_Set: segue os ponteiros para os pais até atingir a raiz.
- Union: faz a raiz de uma árvore apontar para a raiz da outra.

Heurísticas para Florestas de Conjuntos Disjuntos

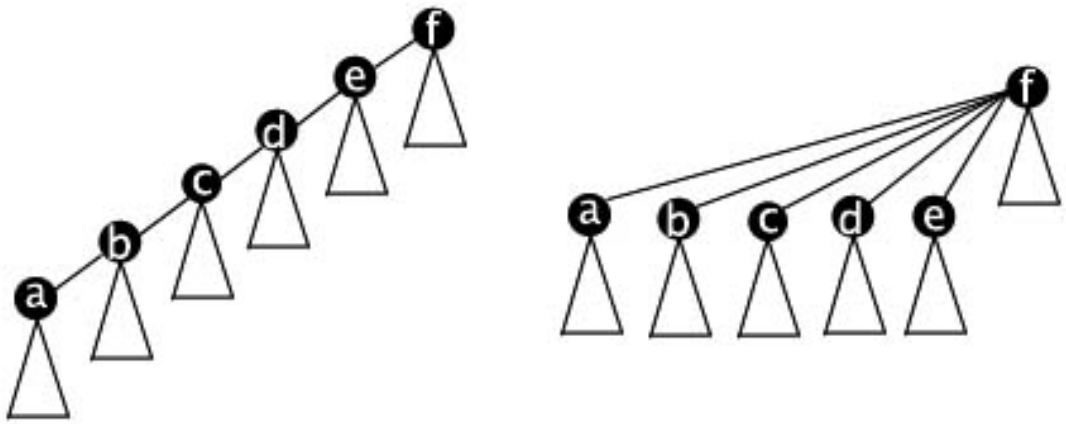
Heurísticas para Florestas de Conjuntos Disjuntos

Union By Rank (União por Ordenação): faça a raiz da árvore com menor número de elementos apontar para a raiz da árvore com maior número de elementos

Heurísticas para Florestas de Conjuntos Disjuntos

Union By Rank (União por Ordenação): faça a raiz da árvore com menor número de elementos apontar para a raiz da árvore com maior número de elementos

Path Compression (Compressão de Caminhos): durante uma busca faça os nós do caminho apontar para a raiz



O Efeito das Heurísticas

Observações:

O Efeito das Heurísticas

Observações:

- Separadamente, ambas as heurísticas melhoram o tempo de execução das operações

O Efeito das Heurísticas

Observações:

- Separadamente, ambas as heurísticas melhoram o tempo de execução das operações
- Sozinha, a heurística Union by Rank induz o mesmo tempo de execução da heurística de união ponderada
  - A implementação executa em tempo O(m log n)

O Efeito das Heurísticas

Observações:

- Separadamente, ambas as heurísticas melhoram o tempo de execução das operações
- Sozinha, a heurística Union by Rank induz o mesmo tempo de execução da heurística de união ponderada
  - A implementação executa em tempo O(m log n)
- Sozinha, a heurística de compressão de caminho induz tempo de execução $\theta(n + f\ log\ n)\ \ para\ \ f\ <\ n$ e onde f é o número de operações Find_Set

O Efeito das Heurísticas

Observações continuação:

O Efeito das Heurísticas

Observações continuação:

- Quando aplicamos ambas as heurísticas, o tempo de execução se torna $O(m\ \alpha\ (m, n))$, onde $\alpha\ (m, n)$ é uma função que cresce muito lentamente, é o inverso da função de Ackermann. Podemos assumir que $\alpha\ (m, n) \leq 4$, para todos os fins práticos $(m, n < 10^{80})$

O Efeito das Heurísticas

Observações continuação:

- Quando aplicamos ambas as heurísticas, o tempo de execução se torna $O(m\ \alpha\ (m, n))$, onde $\alpha\ (m, n)$ é uma função que cresce muito lentamente, é o inverso da função de Ackermann. Podemos assumir que $\alpha\ (m, n) \leq 4$, para todos os fins práticos $(m, n < 10^{80})$

- Kruskal: $O(m\ log\ m + n.Union + m.Find) \Rightarrow O(m\ log\ m + n\alpha(m, n) + m\alpha(m, n)) = O(m\ log\ m)$

Fim

Fim

- Muito Obrigado

Fim

- Muito Obrigado
- Perguntas?

Colômbia


www.turismocolombia.com
www.ingeominas.gov.co
FONDO DE PROMOCIÓN TURÍSTICA
COLOMBIA
Mar Caribe
Región del Caribe
Venezuela
Región del Pacífico
Región Andina
Región de la Orinoquia
Océano Pacífico
Ecuador
Región de la Amazonia
Brasil
Perú

Referências

Referências

- Cormen T. Leiserson C. Rivest R. Stein C. Algoritmos - Teoria e Prática. Tradução da 2^a edição americana, Editora Campus,2002.

Referências

- Cormen T. Leiserson C. Rivest R. Stein C. Algoritmos - Teoria e Prática. Tradução da 2^a edição americana, Editora Campus,2002.
- Artur Lira dos Santos e outros, Conjuntos e Ordenação, IF672 - Algoritmos e Estruturas de Dados CIn - UFPE, 2005. Disponível em: *moreno.cin.ufpe.br/if672/2007.1/monitoria/aula6/* arquivo Conjuntos_Ordenacao.ppt

Referências

- Cormen T. Leiserson C. Rivest R. Stein C. Algoritmos - Teoria e Prática. Tradução da 2^a edição americana, Editora Campus,2002.
- Artur Lira dos Santos e outros, Conjuntos e Ordenação, IF672 - Algoritmos e Estruturas de Dados CIn - UFPE, 2005. Disponível em: *moreno.cin.ufpe.br/if672/2007.1/monitoria/aula6/* arquivo Conjuntos_Ordenacao.ppt
- Eduardo Camponogara,Conjuntos Disjuntos,Departamento de Automaçao e Sistemas Universidade Federal de Santa Catarina, 25 de abril de 2007. Disponível em: *www.das.ufsc.br/camponog/Disciplinas/DAS − 9003/slides_CLR/* arquivo L12.pdf.