

# Indexação Espacial

Lubia Vinhas

# Métodos de acesso

- Métodos de acesso são os procedimentos empregados pelo gerenciador de banco de dados com o objetivo de acelerar a localização e a recuperação de algum dado
- Em BD convencionais, a criação de índices é opcional, ficando a cargo do administrador do banco de dados
- Cada índice provê uma maneira rápida de buscar dados dentro de uma tabela
  - Existem diferentes tipos de índices, cada qual adequado a um tipo de atributo ou situação

# Recuperação de Informações em BD

- Uma consulta típica: quais são as informações referentes ao aluno de matrícula **100203**

Tabela: Alunos				
Nº Reg	Matricula	Nome	Telefone	Data Matrícula
1	9765421	Gilberto	11-2345	1969
2	8763524	Antônio	12-3456	2000
	1000203	Maria	10-0987	2004
	6827265	José	08-9865	2005
5	2039457	João	23-8754	2002
6	1000204	Eduardo	23-5677	1998
7	8888823	Ana	55-5676	1997
8	5678999	Carolina	58-5676	1943
...	...	...	...	
100000	1243547	Cássia	56-5676	2000

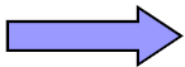
Em uma busca sequencial serão feitas 3 comparações

# Recuperação de Informações em BD

- Uma consulta típica: quais são as informações referentes ao aluno de matrícula **1243547**

Tabela: Alunos				
Nº Reg	Matricula	Nome	Telefone	Data Matrícula
1	9765421	G ilberto	11-2345	1969
2	8763524	A ntônio	12-3456	2000
3	1000203	M aria	10-0987	2004
4	6827265	José	08-9865	2005
5	2039457	João	23-8754	2002
6	1000204	Eduardo	23-5677	1998
7	8888823	A na	55-5676	1997
8	5678999	C arolina	58-5676	1943
...	...	...	...	
100000	1243547	C ássia	56-5676	2000

Em uma busca sequencial serão feitas 100000



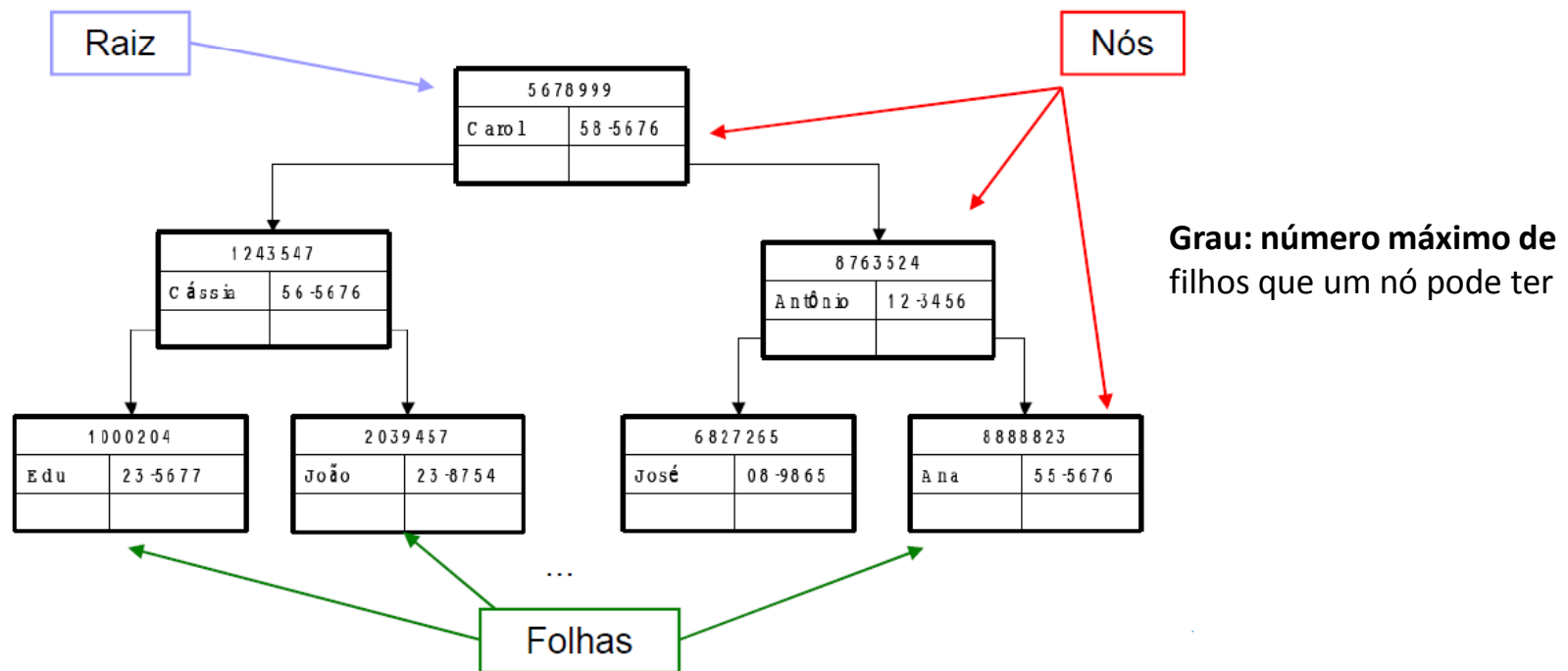
# Recuperação de Informações em BD

- Um algoritmo de busca sequencial tem complexidade linear:  $O(n)$

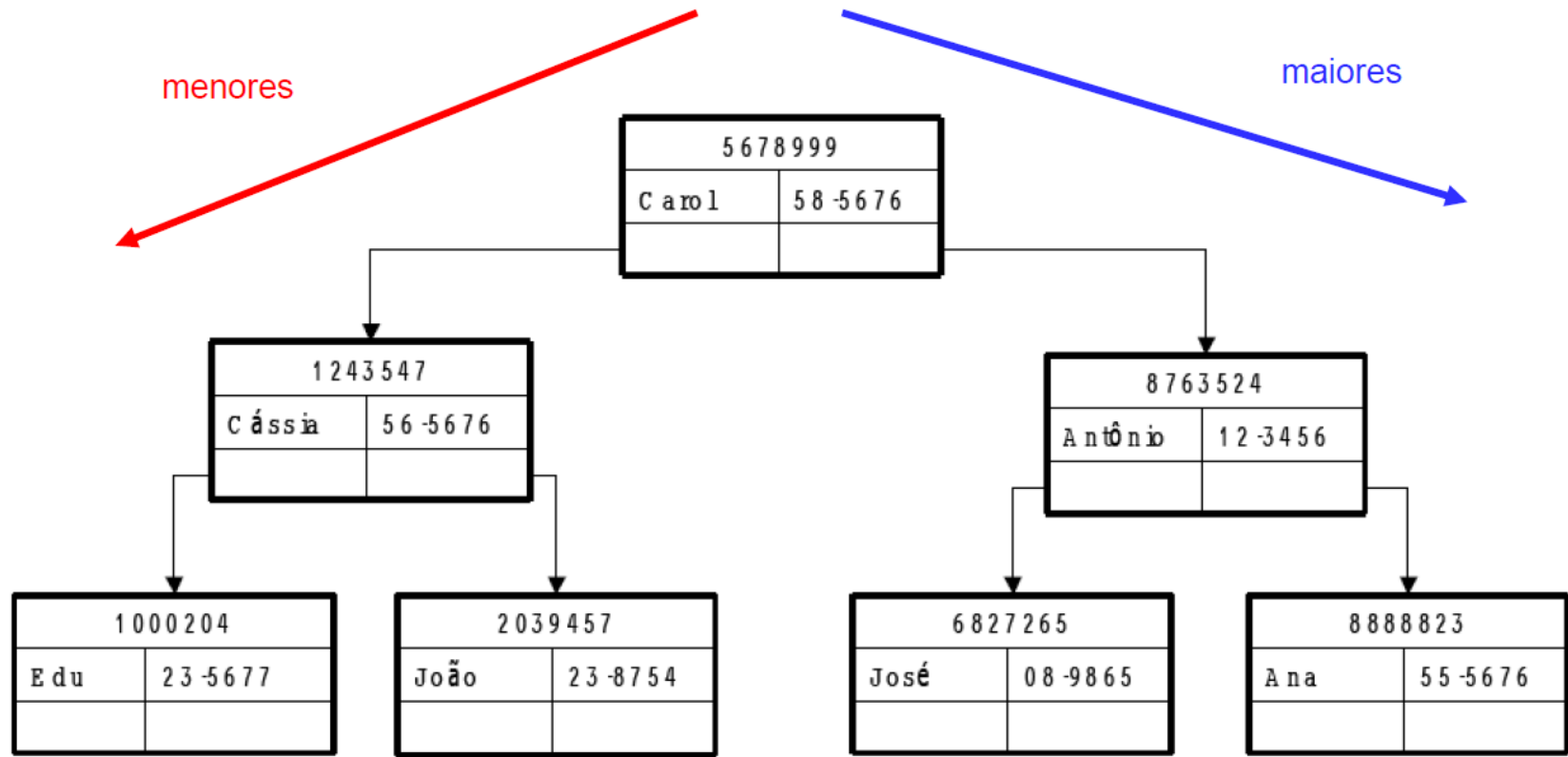


# Índices

- Estrutura de indexação convencional mais comum em bases de dados baseiam-se em estruturas de dados do tipo *árvore*

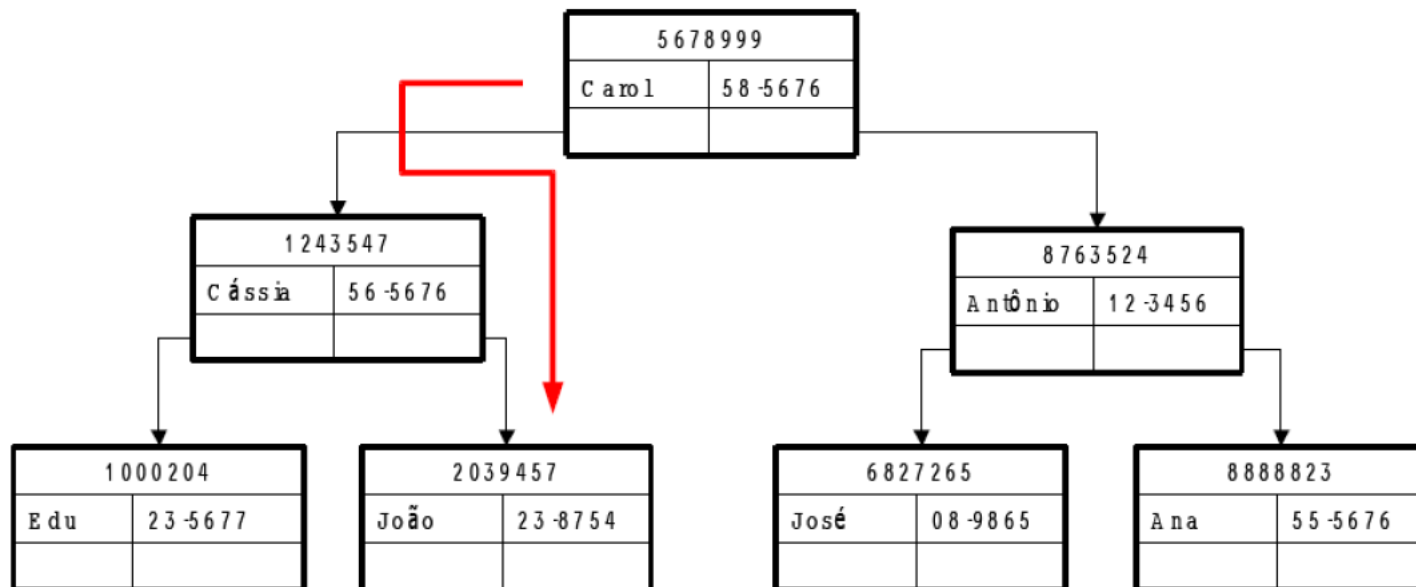


# Árvores Binárias



# Árvores binárias

- Quais os dados do aluno cuja matrícula é 2039457?

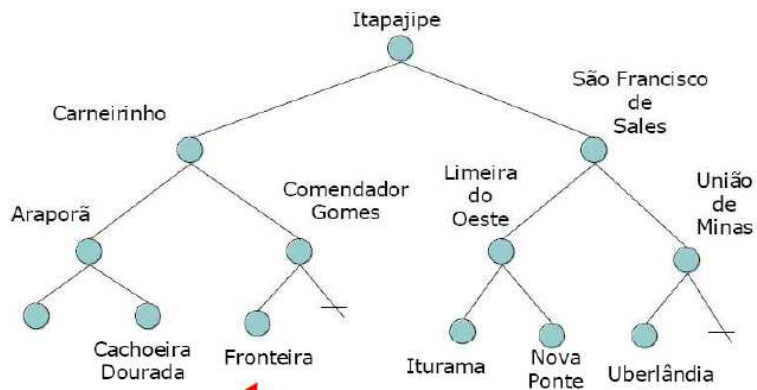




# Árvores binárias de pesquisa (BST)

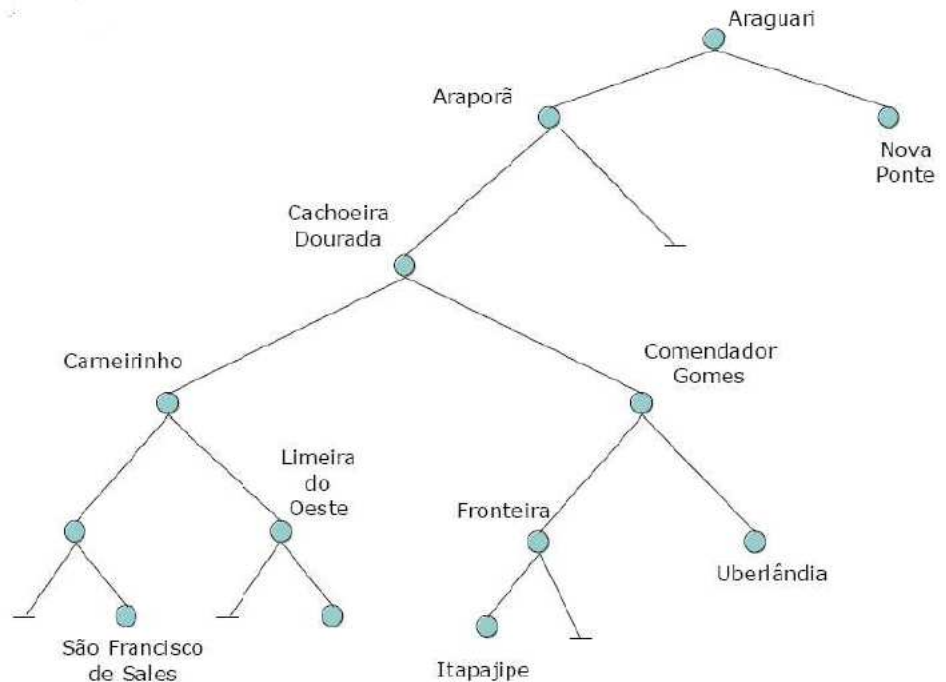
- Grau máximo de um nó: 2
- Árvores balanceadas mais comuns:
  - AVL (Adelson-Velskii e Landis);
  - Red Black Tree;
  - Splay Tree.
- Operações em  $O(\log_2 n)$ 
  - Localizar um nó, dada um valor de chave;
  - Inserir um nó, dado um valor de chave;
  - Remover um nó, dado um valor de chave.

# Árvores Balanceadas



Balanceada

Não-Balanceada



# Métodos de acesso espaciais

- Em BD espaciais, o SGBD precisa contar com métodos de acesso especificamente voltados para as características dos dados de representação
- Os métodos tradicionais são também usados, mas apenas sobre os dados alfanuméricos

# Árvores binárias

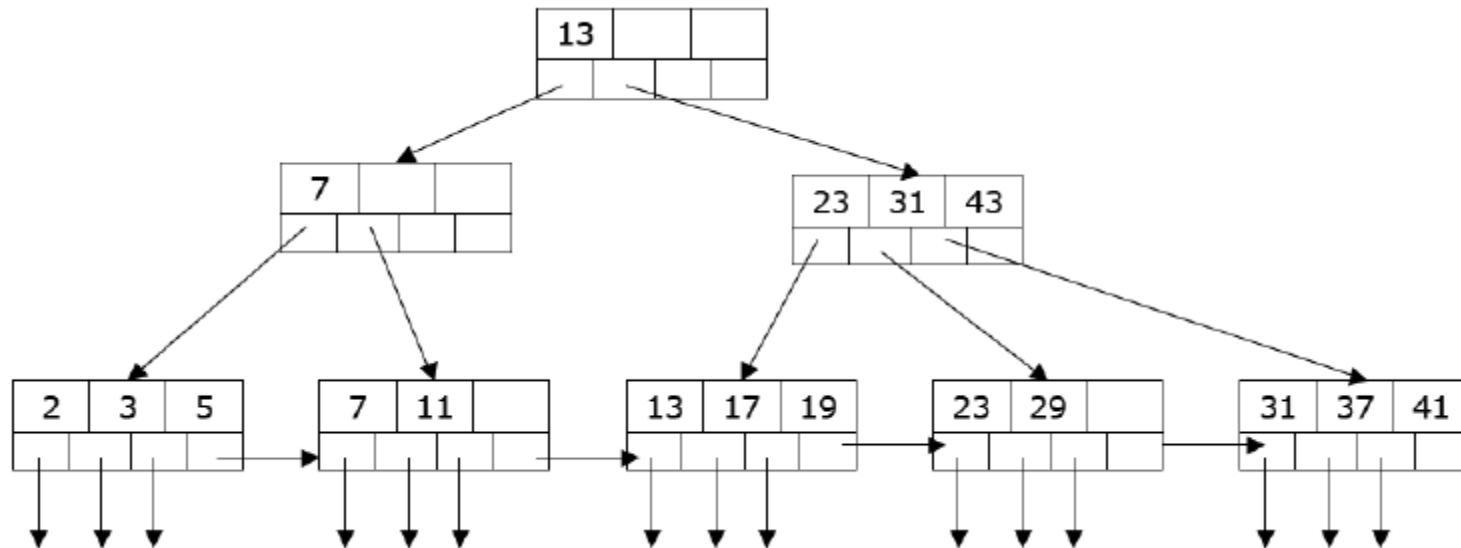
- São usadas principalmente em memória principal.
- No caso de grandes bancos de dados a memória principal pode não ser suficiente para armazenar todos os nós da árvore:
- Comum armazená-las em disco;
- Nesse caso utilizamos uma representação que procura minimizar o acesso a disco;
- A forma mais comum, e mais largamente empregada pelos sistemas comerciais atuais, é a representação do índice através de uma Árvore-B+ .

# Árvores B (B-Tree)

- São árvores de pesquisa balanceadas especialmente projetadas para a pesquisa de informação em discos magnéticos e outros meios de armazenamento secundário (externos).
- Minimizam o número de operações de movimentação de dados (escrita/leitura) numa pesquisa ou alteração.
- O grau de um nó pode ser alto.
- Podem ser consideradas como uma generalização natural das árvores de pesquisa binárias.

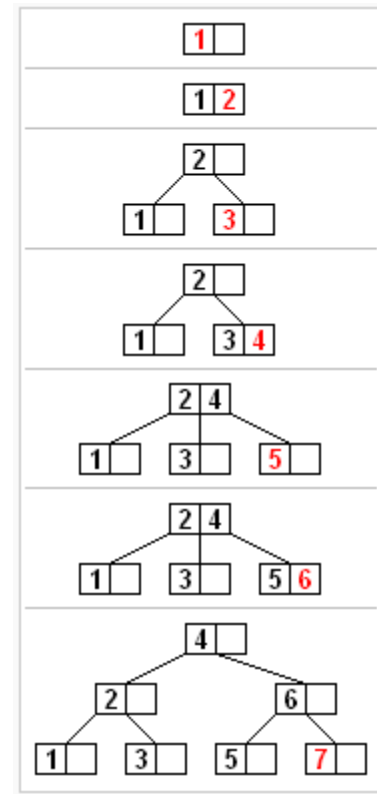
# Árvores B (B-Tree)

- Uma B-tree de ordem  $m$  é tal que:
- Todas as folhas estão no mesmo nível
- Todos os nós internos, exceto a raiz, podem ter no máximo  $m$  e no mínimo  $m / 2$  filhos
- A raiz tem no mínimo 2 e no máximo  $m$  filhos



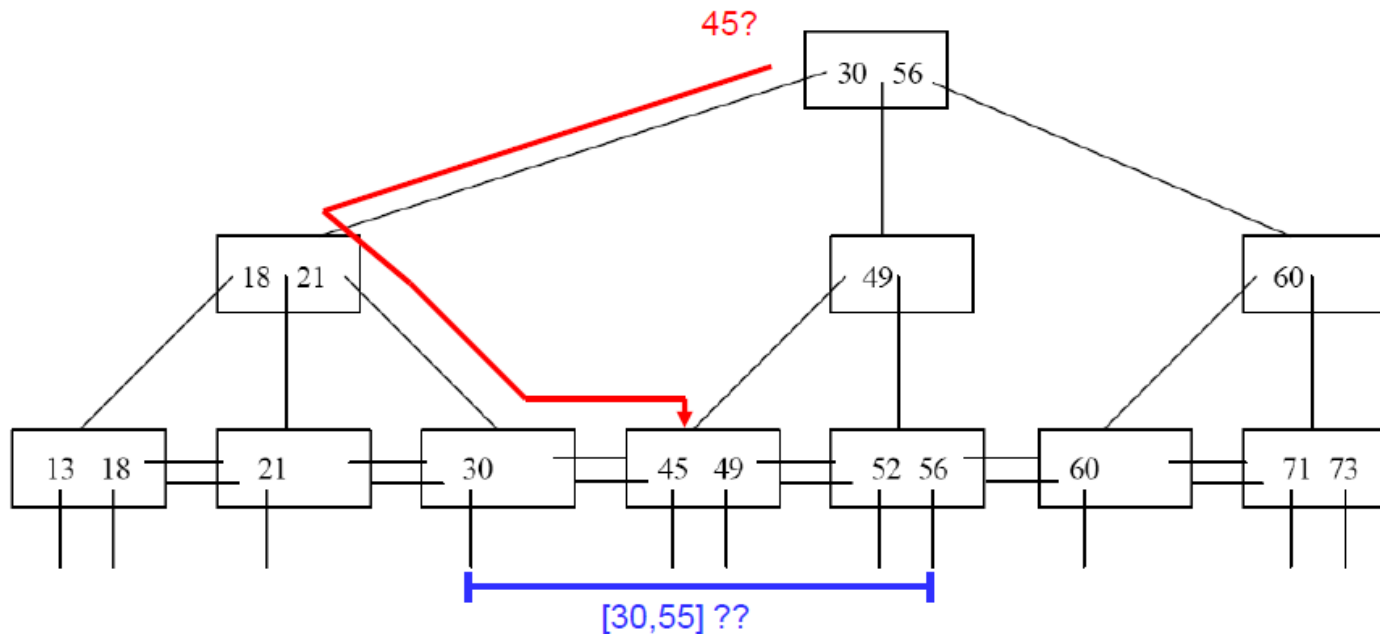
# Árvores B

- Como os número de filhos dos nós internos pode variar, e esses não necessitam estar cheios, a árvore não necessita ser totalmente rebalanceada sempre.
- Cada nó é armazenado em uma página de disco.
  1. Se o nó contém menos que o número máximo permitido, insira o elemento mantendo a ordem.
  2. Senão, divida o nó em dois:
    1. O valor médio é escolhido entre os elementos do nó e o novo elemento.
    2. Valores menores que o valor médio são colocados a esquerda e maiores a direita
    3. Insira o valor médio no nó acima (caso ele tenha pai) seguindo o mesmo procedimento. Caso não tenha crie um novo nó.



# Árvores B

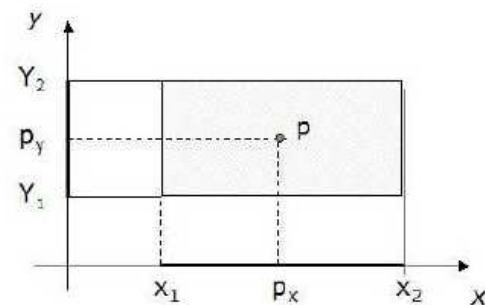
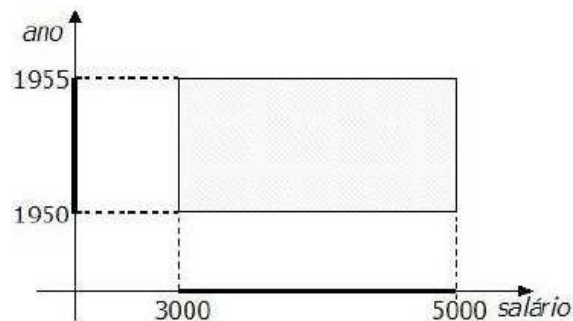
- Todas as chaves são mantidas em folhas, e algumas chaves são repetidas em nós não-folha para definir caminhos para localizar registros individuais
- As folhas são ligadas através de uma lista duplamente encadeada, de modo a oferecer um caminho seqüencial para percorrer as chaves na árvore.





# Índices multidimensionais

- Tanto a Árvore Binária de Pesquisa quanto a Árvore-B+, são estruturas unidimensionais, ou seja, pressupõem que a chave de pesquisa seja formada por apenas um atributo ou pela concatenação de vários atributos
- Facilitam o processamento das consultas por intervalos quando estes são unidimensionais
- Muitas vezes a busca que se deseja fazer é multidimensional



*Quais os empregados com idade entre 50 e 55 anos com salário entre 3000 e 5000?*

# Métodos de acesso espaciais

- Métodos de acesso espaciais são estruturas de dados auxiliares, porém essenciais para o processamento de consultas e para a execução de procedimentos de análise espacial com eficiência
- Também são chamados de índices espaciais  
Ao contrário dos índices convencionais, os espaciais são de uso obrigatório, para que o desempenho seja minimamente aceitável em BDs de tamanho razoável

# Métodos de acesso espaciais

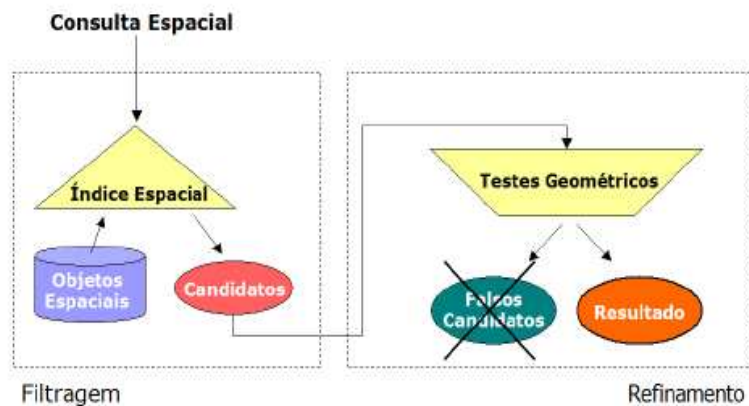
- Em geral, uma consulta envolve apenas uma pequena parcela do BD
- Percorrer todo o BD procurando pelos dados relevantes para a consulta é em geral muito ineficiente
- O método de acesso estabelece um *plano de execução* para a consulta

# Métodos de acesso espacial

- O plano de execução realiza uma *filtragem*, para determinar um subconjunto dos objetos do BD que *podem atender às especificações da consulta*
- Essa filtragem precisa ser executada com muita rapidez, e portanto é realizada sobre uma *aproximação da forma geométrica* de cada objeto

# Índices espaciais

- Busca atender a consultas onde objetos são selecionados com respeito a sua localização espacial
- Índices tradicionais dependem de uma ordem total em uma chave
- Índices espaciais buscam preservar proximidade espacial



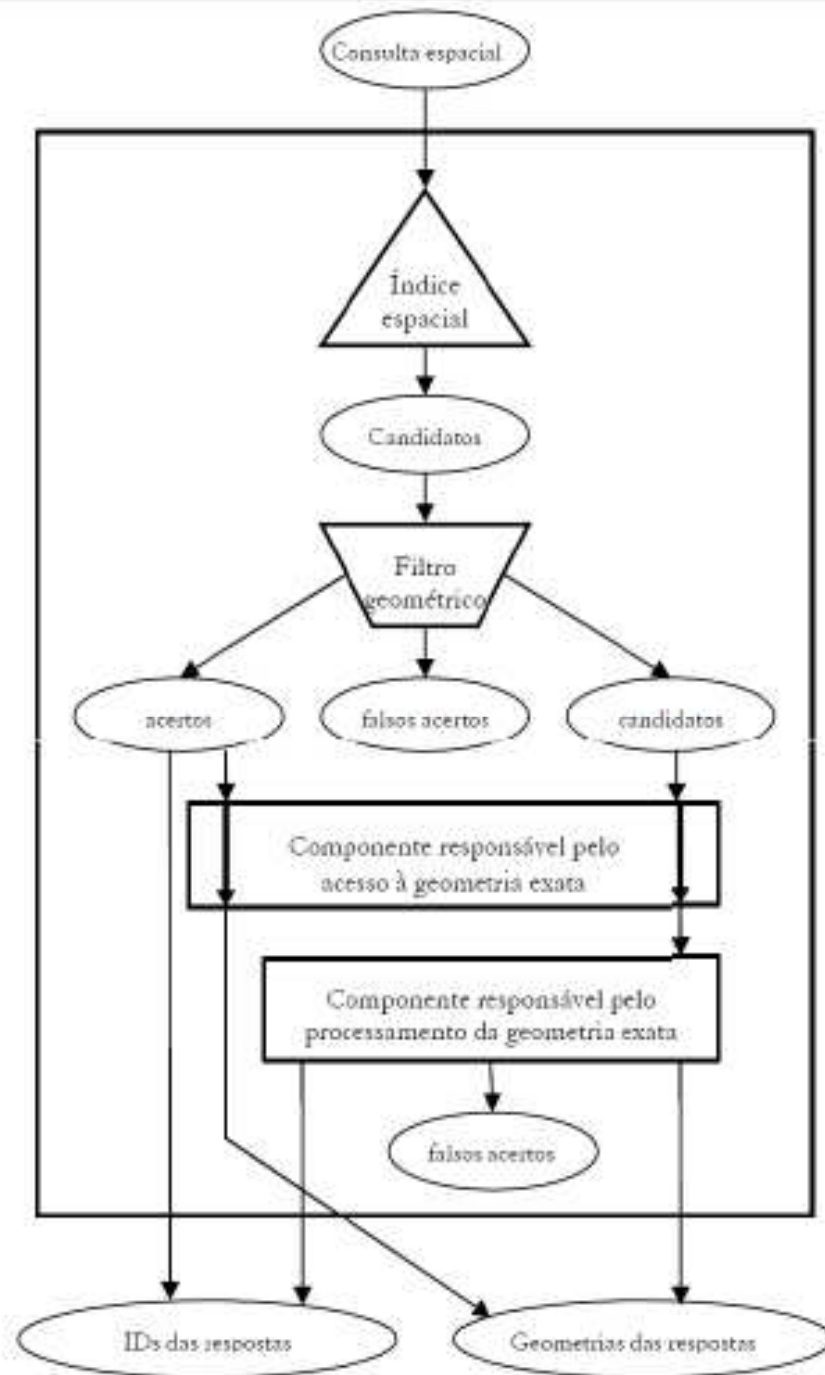
Seleção por ponto



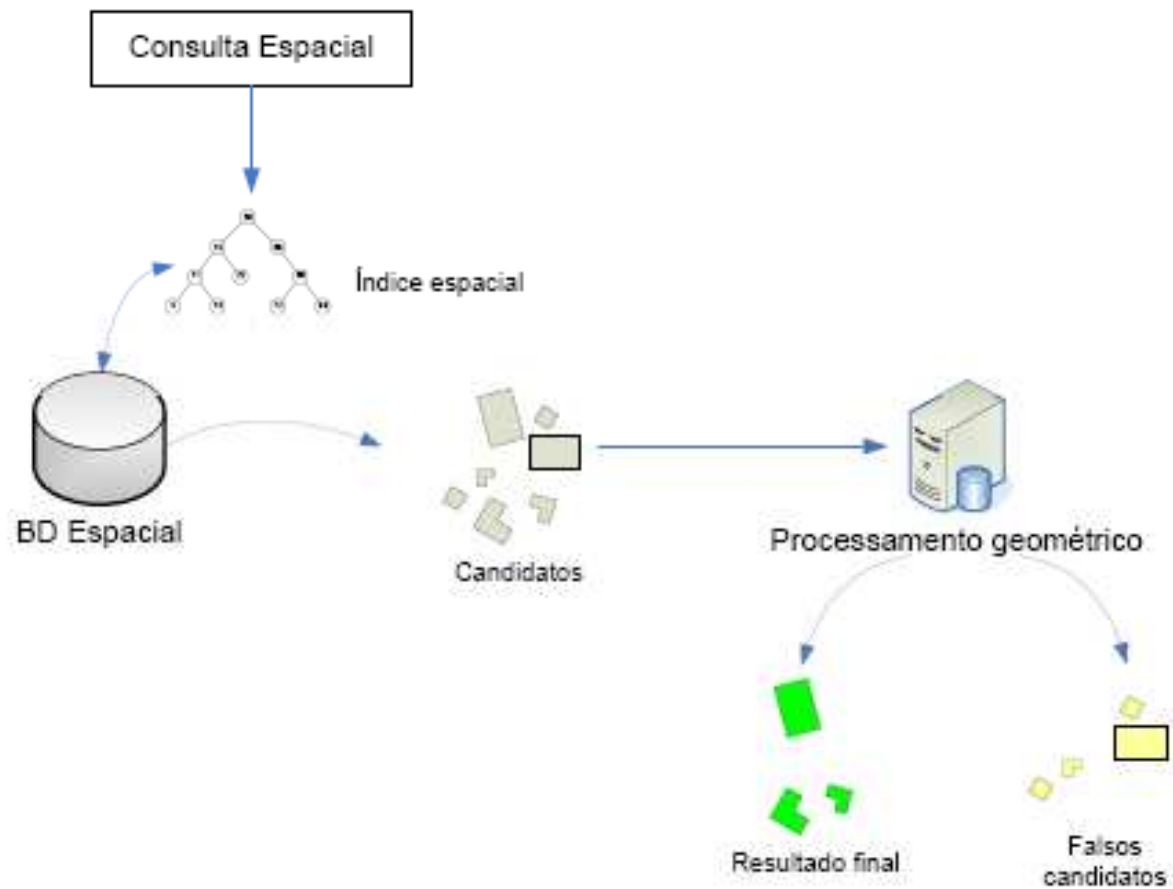
Seleção por região



Seleção por janela



# Processamento de consulta espacial



Clodoveu Davis

# Uso de índices espaciais

- Seleção de objetos para visualização
  - Objetos contidos no retângulo do zoom
- Localização de objetos selecionados por
  - apontamento
  - Objetos cujas fronteiras contêm ou se aproximam do ponto indicado na tela
- Consultas topológicas
  - Encontrar objetos relacionados topologicamente a um objeto espacial dado (contido em, contém, adjacente a, cruzando, etc.)



# Métodos de indexação espacial

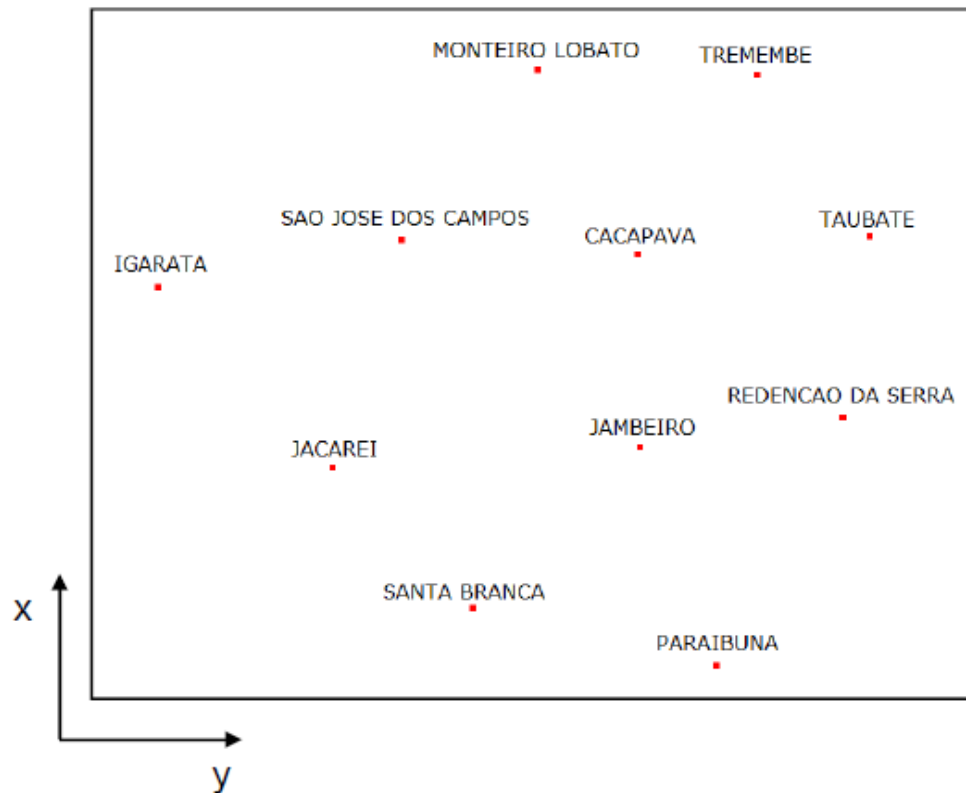
- **Dirigidos pelo espaço:** baseiam-se em partições do espaço independente da distribuição dos dados (pontos ou retângulo envolvente) no plano 2D.
- **Dirigidos pelos dados:** particionam um conjunto de objetos e não o espaço

# K-d Tree

- Indexa chaves formadas por *k atributos* (geometricamente, dimensões)
- Cada nível da árvore corresponde a uma das dimensões
- As dimensões ocorrem ciclicamente pelos níveis da árvore

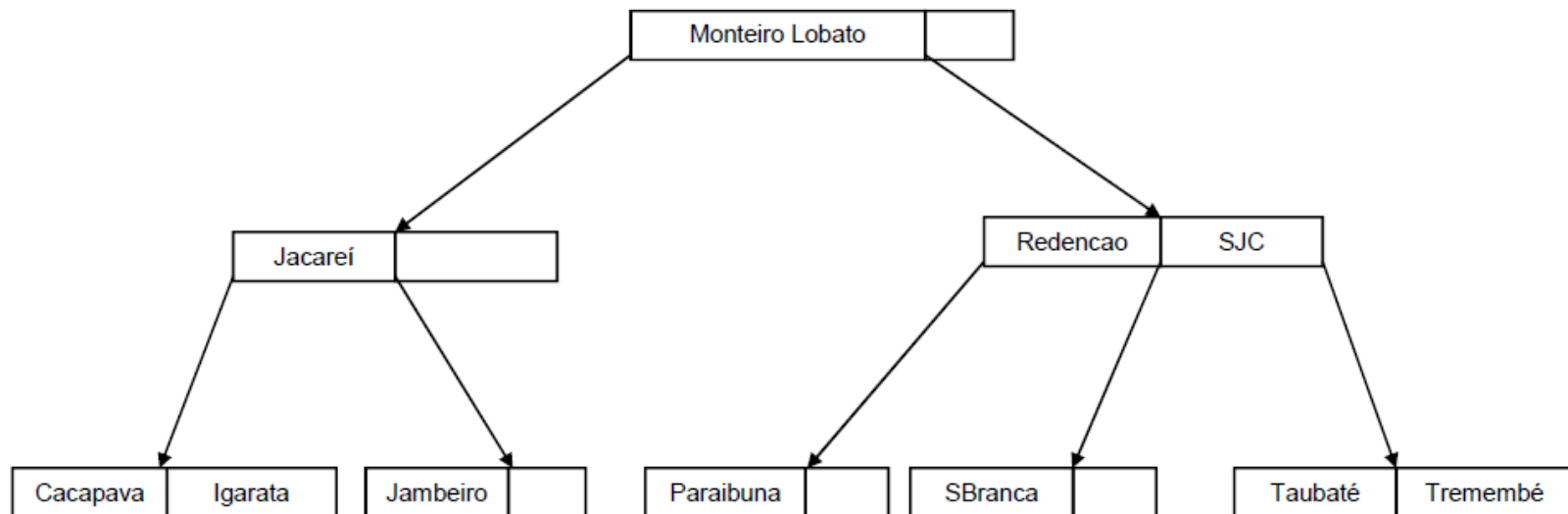
# K-d Tree

- Generalização árvore binária de pesquisa para o espaço multidimensional. Ex: localização das cidades do Vale do Paraíba



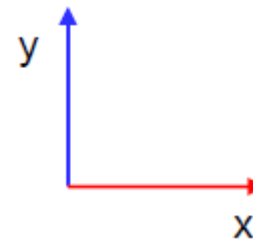
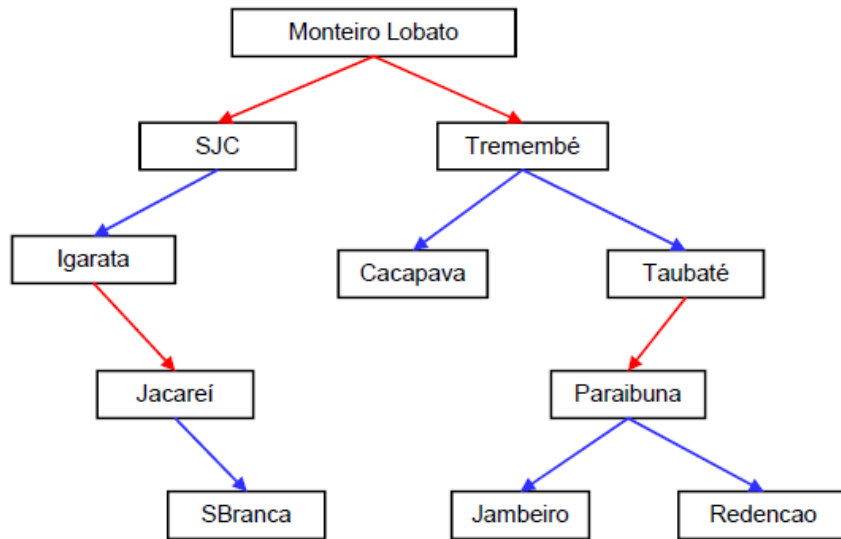
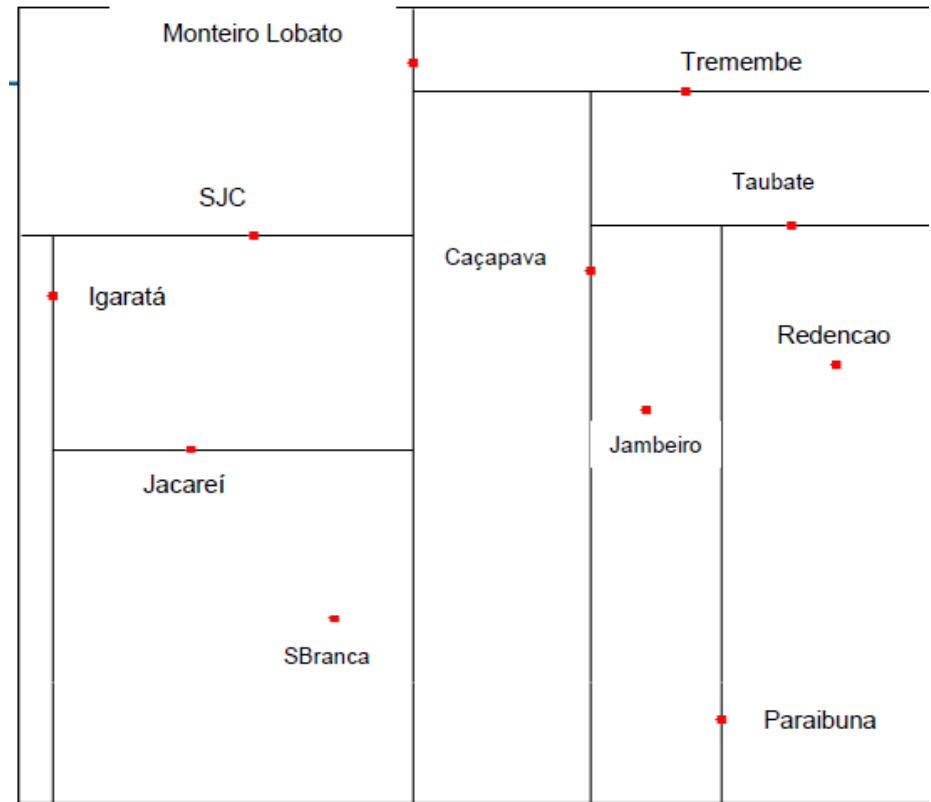
# K-d Tree

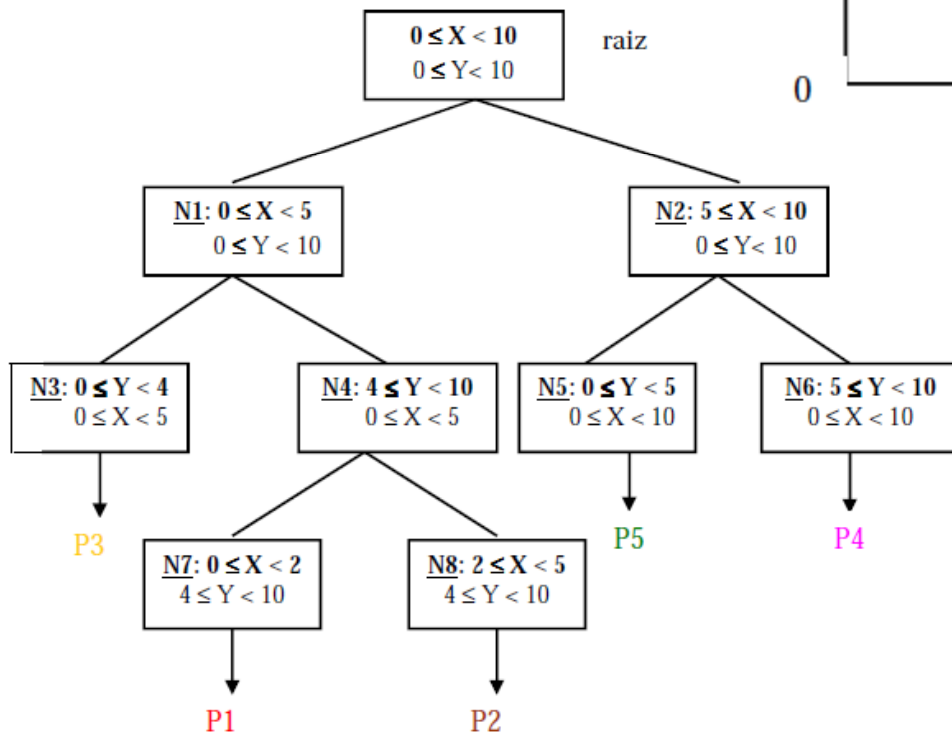
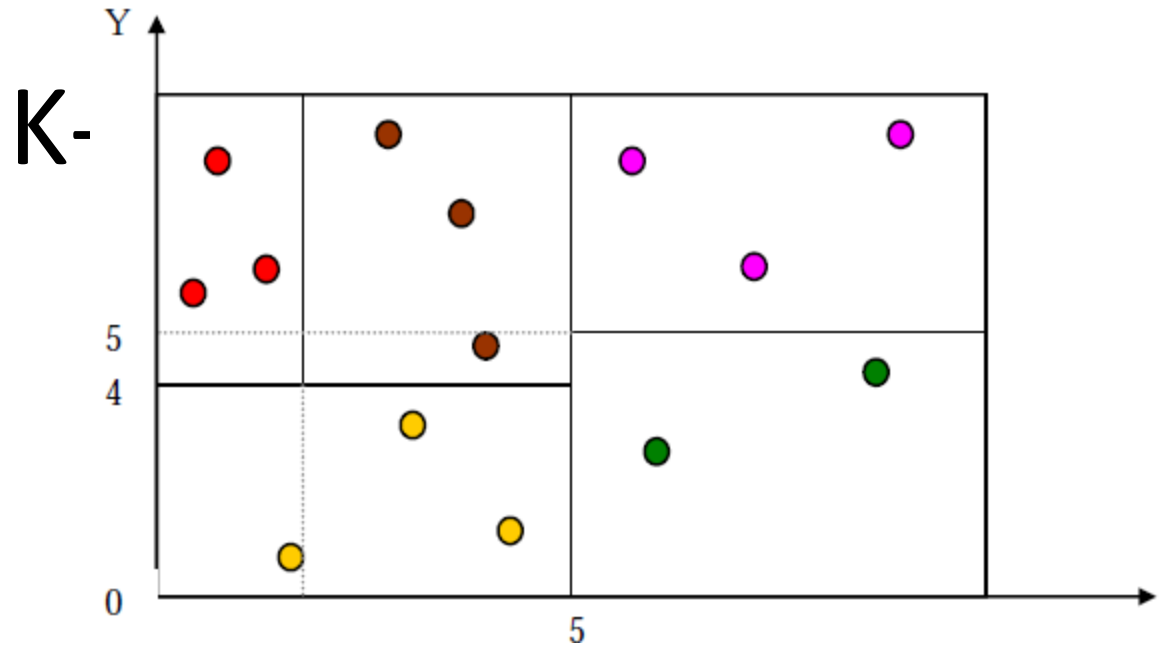
- B-Tree sobre os nomes. Responde bem a pergunta: *“encontre a cidade de Paraibuna”*, mas não a pergunta *“encontre a cidade que intercepta a janela X”*



# K-d

Exemplo de k-d Tree com 11 pontos





- Para localizar o ponto (4,9): raiz, N1, N4, N8 até a página P2.
- Para localizar os pontos no retângulo ((1,4),(4,9)): raiz, N1, N4, N7 e N8 até as páginas P1 e P2.

# K-d Tree: Busca

- Seja um retângulo  $R=[(x1, y1): (x2, y2)]$  contendo o intervalo de pesquisa:
  - Começamos pela raiz (nível 0  $\rightarrow$  par)
  - Se o ponto deste nó  $\in R$ , ele é reportado
  - Se ele for um nó em um nível par:
    - Se  $R.x1 < p.x$ , aplicamos recursivamente o passo 2 à sub-árvore esquerda
    - Se  $R.x2 > p.x$ , aplicamos recursivamente o passo 2 à sub-árvore direita
  - Caso contrário, se ele for um nó em um nível ímpar:
    - Se  $R.y1 < p.y$ , aplicamos recursivamente o passo 2 à sub-árvore esquerda
    - Se  $R.y2 > p.y$ , aplicamos recursivamente o passo 2 à sub-árvore direita
- Pesquisa por apontamento:  $O(\log_2 n)$
- Pesquisa por janela:  $O(N)$

# K-d Tree

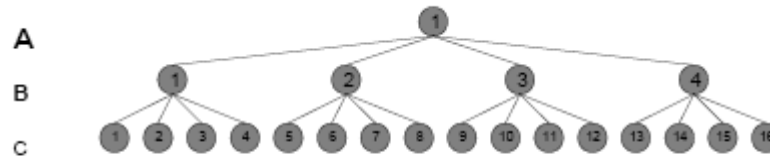
- Estrutura de dados  $d$ -dimensional que representa a subdivisão recursiva do espaço em subespaços por meio de  $d-1$  hiperplanos. “ $k$ ” representa a dimensão do espaço.
- Uma K-d tree é uma árvore de busca binária
- Os hiperplanos são orientados alternadamente entre as  $d$  possibilidades
- Cada particionamento do plano contém pelo menos um ponto, que é usado para ser representado na árvore.
- Inserções e buscas são simples, mas remoções não.
- Existem muitas variações e extensões da K-d tree: k-d tree adaptativa, hB-Tree, QuadrTree, k-b-B Tree



# Quad-Tree

- Espécie de árvore em que cada nó possui sempre quatro “folhas”
- A cada nó corresponde uma região quadrada do espaço
- Os objetos são relacionados ao menor quadrado que contém seu retângulo envolvente

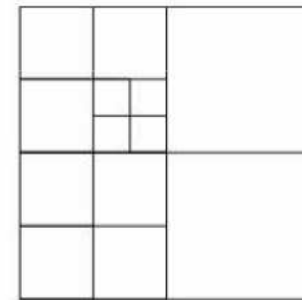
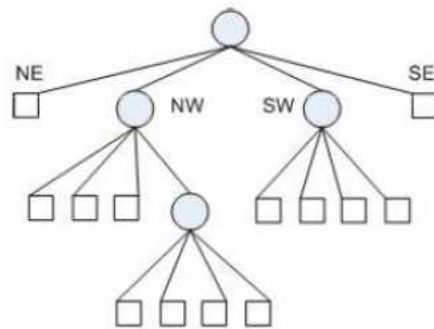
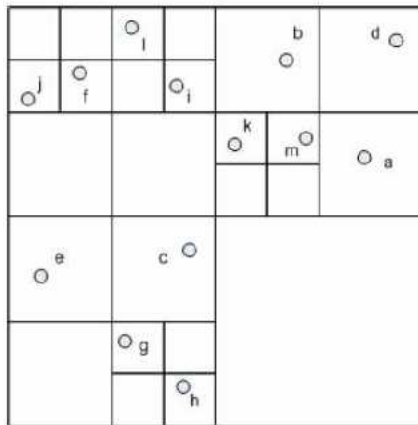
# Quad-Tree



C1	C2	C3	C4
B1		B2	
C5	C6	C7	C8
A1			
C9	C10	C11	C12
B3		B4	
C13	C14	C15	C16

# QuadTrees

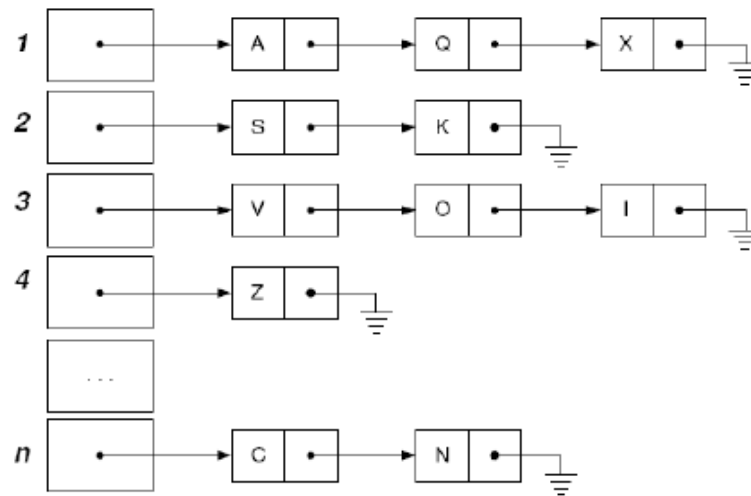
- Baseia-se no conceito de divisão por quadrantes
- O espaço é particionado até que a capacidade da página seja atingida



QuadTree de pontos

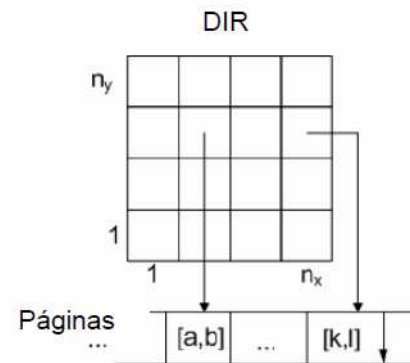
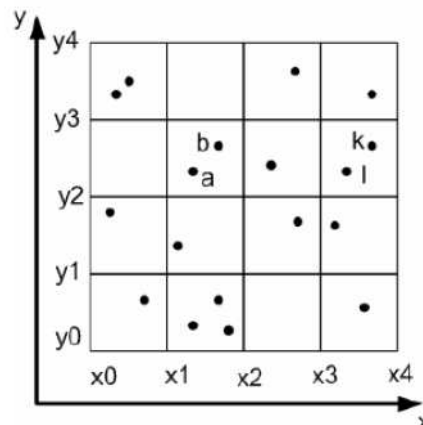
# Hashing

- Transformação de chave: consiste em criar uma série de *pacotes*, numerados seqüencialmente que receberão os identificadores
- Cada identificador que chega, seja para ser inserido, seja para ser pesquisado, é transformado em um número de 1 a n, identificando o *pacotes* correspondente a ele.



# Grades fixas

- O espaço é dividido em uma grade de  $n_x \times n_y$  células, igualmente espaçadas. Cada célula corresponde a uma página de disco
- Um ponto  $P$  é associado a célula  $c$  se ela o contém
- O índice requer uma matriz  $[1:n_x, 1:n_y]$  como um diretório. O elemento  $DIR[i, j]$  contém o endereço de página ID que armazena os pontos associados a célula  $c_{i,j}$

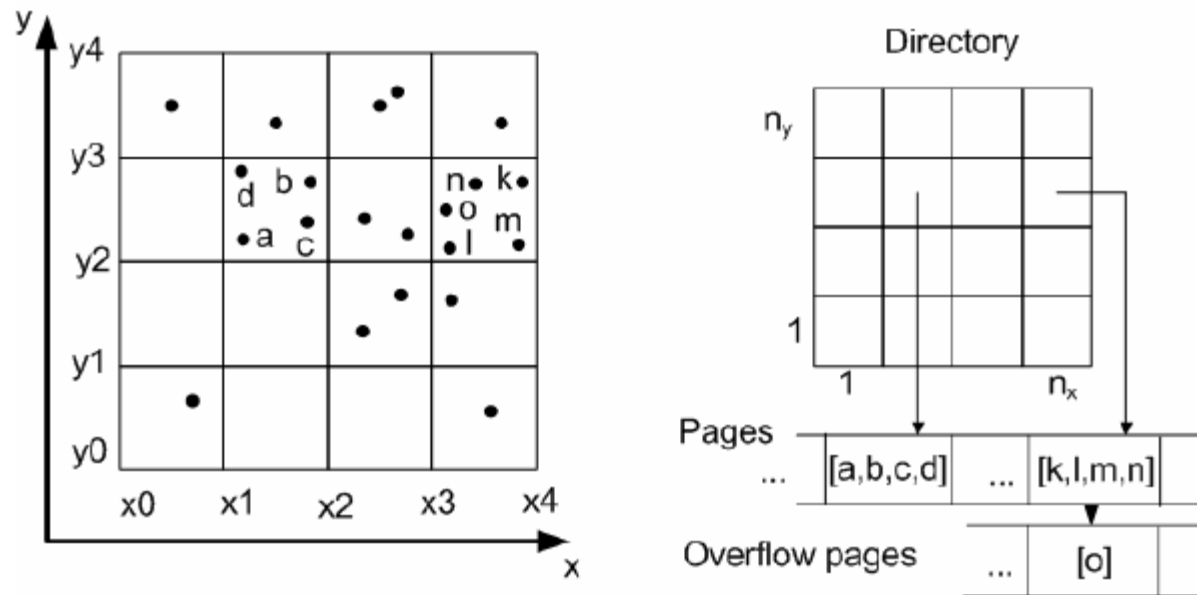


# Grades Fixas

- Se  $[S_x \times S_y]$  é o tamanho do espaço de busca 2D, cada célula tem o tamanho  $[S_x/n_x \times S_y/n_y]$ .
- Operações:
  - Inserindo  $P(a,b)$ :
    - Calcule  $i = (a-x_0)/(S_x/n_x) + 1$  e  $j = (b-y_0)/(S_y/n_y) + 1$
    - Leia a página  $DIR[i,j].Id$  e insira  $P$
  - Consulta por apontamento: dado um ponto  $P(a,b)$  busque a página para inserção, leia a página, passe pelas entradas e verifique qual delas é igual a  $P$ .
  - Consulta por janela: calcule o conjunto de  $S$  de células  $c$  tais que  $c.MBR$  intercepte a janela de consulta  $W$ ; para cada célula  $c_{i,j}$  em  $S$ , leia a página  $DIR[i,j].Id$  e retorne os pontos na página que estão contidos em  $W$
- Consultas por ponto requerem uma operação de E/S
- Número de operações de E/S para executar uma consulta por janela depende do número de intersecções com a janela  $W$ .

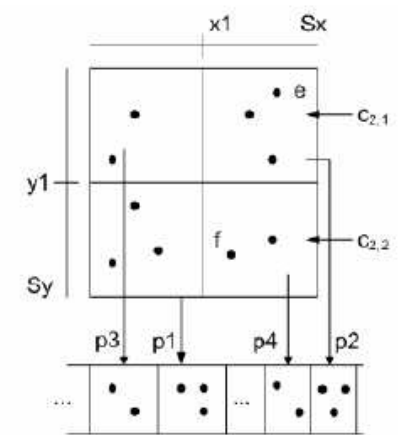
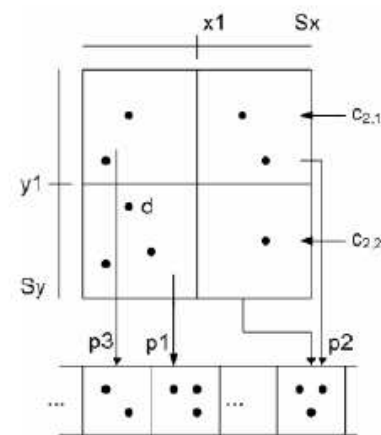
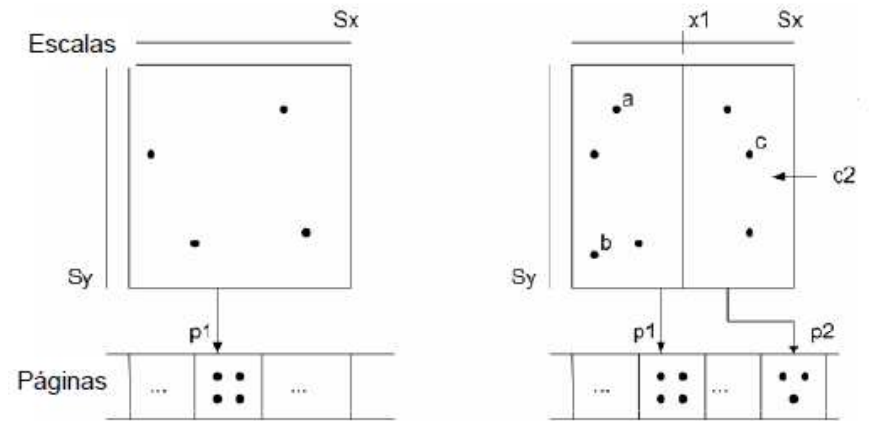
# Grades fixas

- A resolução da grade depende do número de pontos sendo indexado; dada uma capacidade de células  $M$ , é possível criar grades fixas com pelo menos  $N/M$  células.
- Grades fixas são muito sensíveis a distribuição dos pontos



# Grid-files

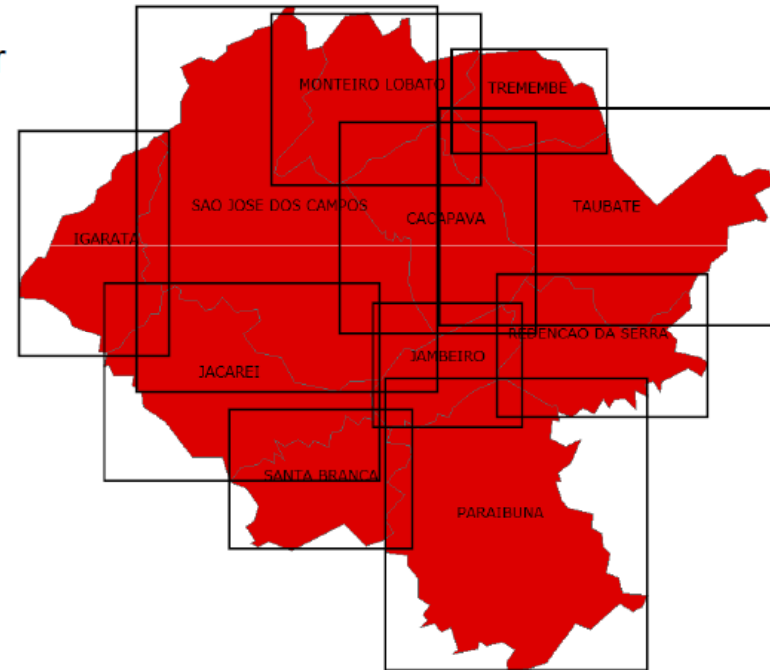
- Partição se adapta a distribuição dos pontos
- No diretório duas células podem apontar para a mesma página
- Duas estruturas que representam a escala de particionamento em cada eixo:  $S_x$  e  $S_y$ .





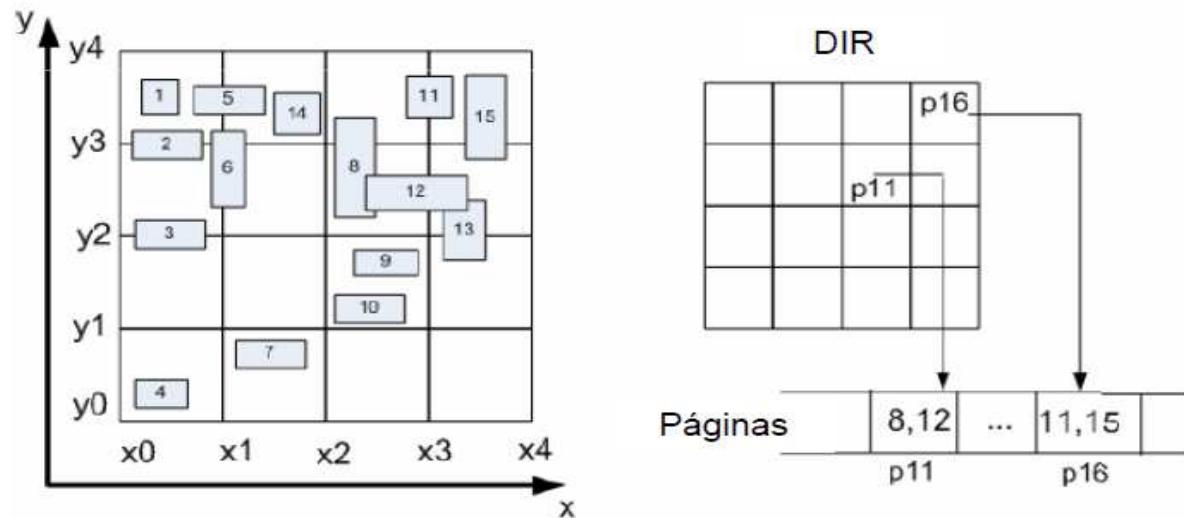
# Retângulo Envolvente

- Para objetos com área adota-se uma aproximação pelo seu menor retângulo envolvente: MBR (*Minimum Bounding Rectangle*)



# Grid-Files

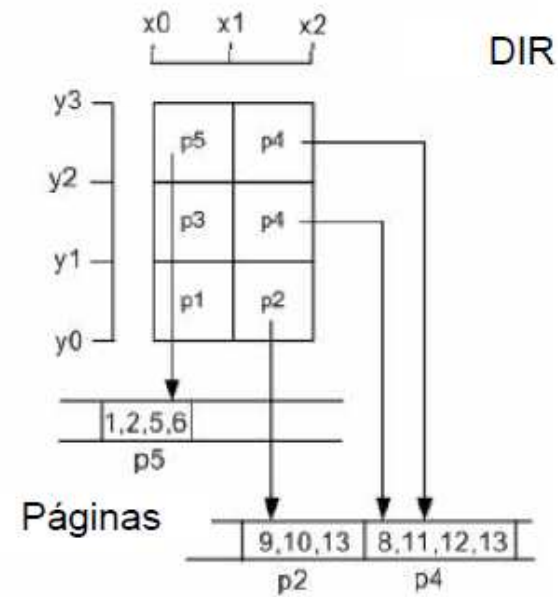
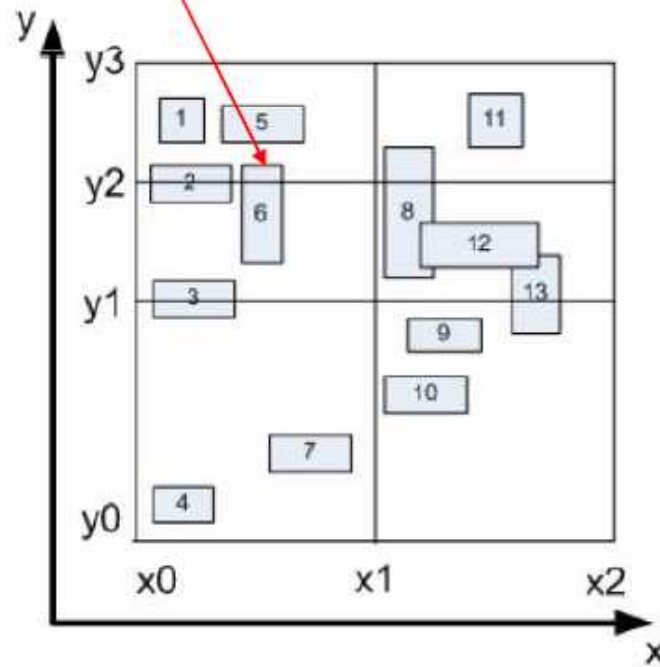
- Objetos são associados as células que seus MBR interceptam



# Grid Files

- Capacidade 4 nas páginas

Causa a divisão

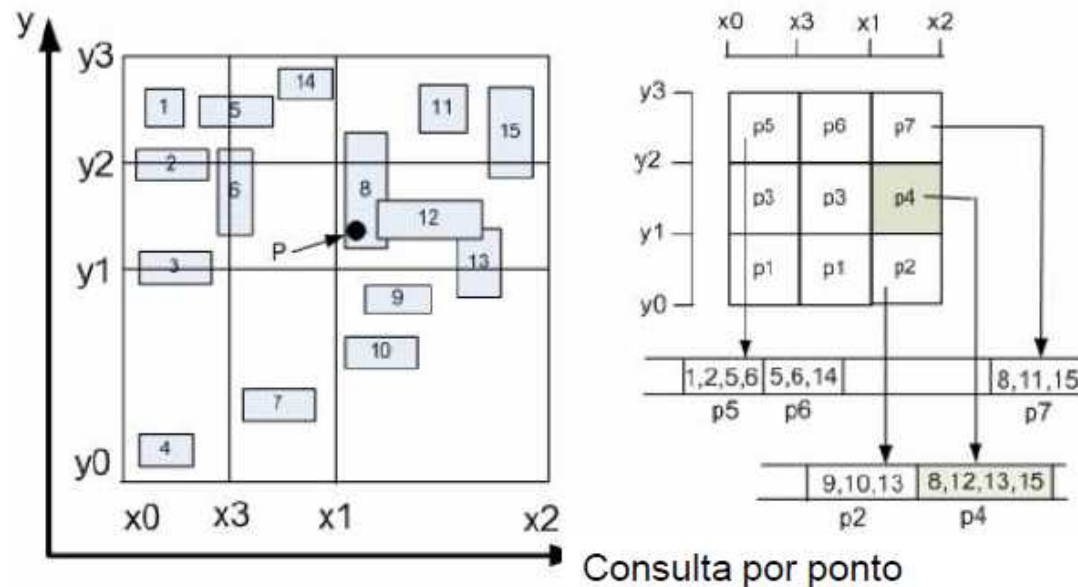


DIR

Páginas

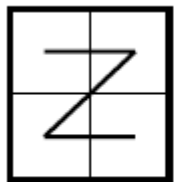
# Grid-Files - consultas

- Dado um ponto  $P(a,b)$ , determine a célula que o contém. Acesse a página correspondente e obtenha a coleção de objetos  $E$  tais que  $P \in e.mbr$ .
- Teste exatamente quais geometrias contém o ponto
- Dado uma janela determine todas as células que a interceptam. Acesse a coleção de objetos, remova as duplicações e teste os objetos.

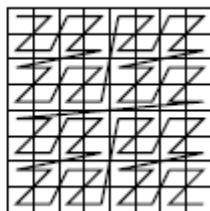
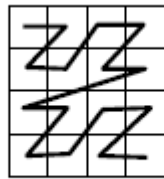


# Space filling curves

- Uma space-filling curves define uma ordem total nas células de uma grade 2-dimensional. A a cada célula é associado um número de forma que números próximos estão associados a células próximas no espaço



z-ordem



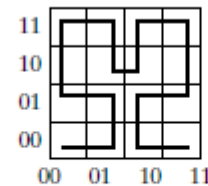
n=0



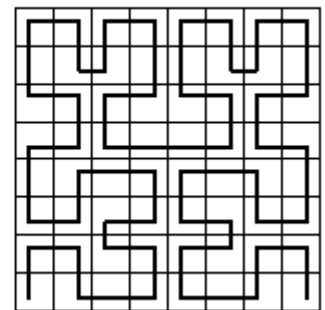
n=1



n=2



n=3



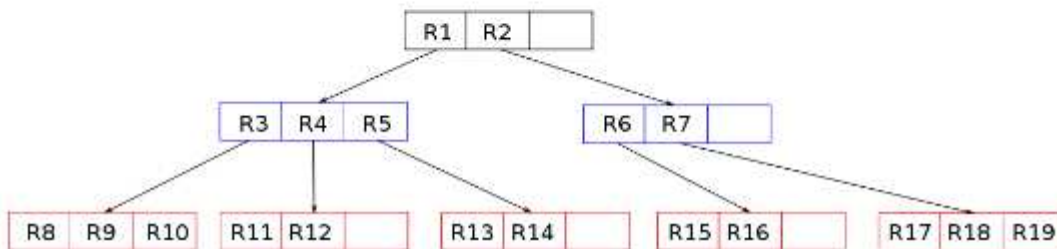
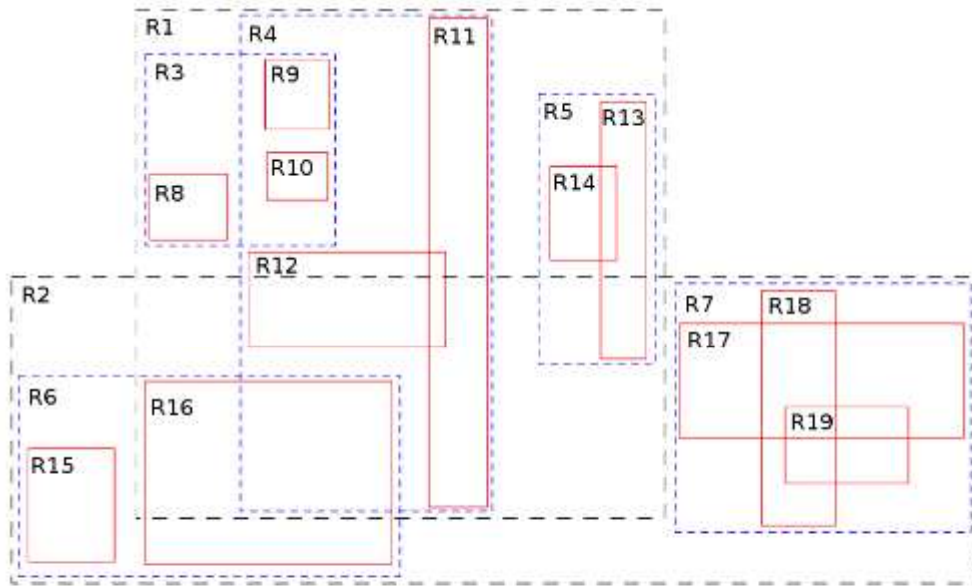
Hilbert

# R-Tree

- Indexa objetos pelo seu retângulo envolvente mínimo
- Pontos também podem ser indexados (retângulo envolvente nulo)
- Cada bloco de armazenamento pode conter um número variável de retângulos
- O aumento do número de objetos causa subdivisões nos blocos, e a redução provoca fusões de blocos

# R-Tree

ff



# R-Tree

- Existem diversas variações na literatura
  - R+-Tree
  - R\*-Tree
  - Hilbert R-Tree
  - X-Tree
  - ...
- O mais usual é encontrar implementações da R-tree original nos produtos



# SAM

- Métodos de acesso espacial são implementados em extensões espaciais e são usados para tornar eficiente a consulta ao dado espacial:
- Oracle spatial: R-Tree, Quad-Tree
- PostGIS: R-Tree-over-GiST
- MySQL: R-Tree