

Estruturas de Dados II

k-d Tree

k-d Tree

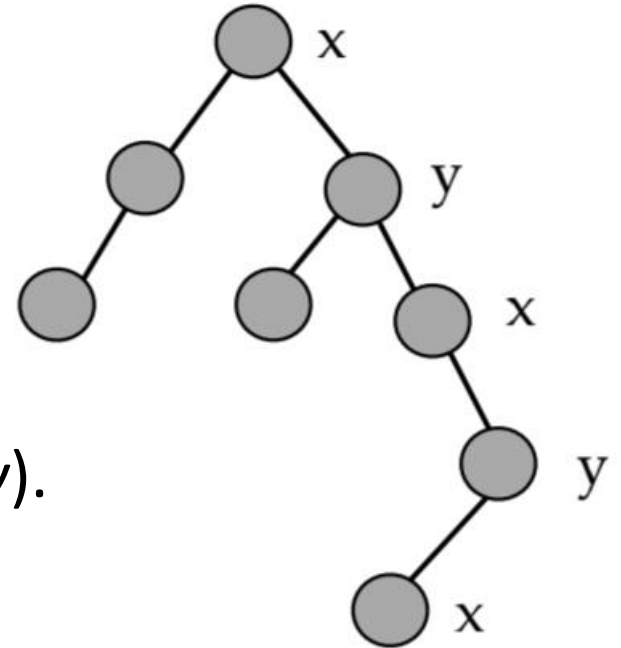
- *k*-d Tree (*k-dimensional* Tree), inventada por Bentley (1975) é uma Árvore Binária de Busca (ABB) onde cada nodo é um ponto *k-dimensional* no espaço. A árvore *k*-d difere da ABB onde cada nível da árvore *k*-d se ramifica baseada numa pesquisa de chave para o nível, chamado discriminador.
- Definimos o discriminador no nível α como sendo $\alpha \bmod k$ para k dimensões. Por exemplo, podemos armazenar dados de coordenadas x, y . Neste caso, k é 2 (há duas coordenadas), com a coordenada x definida arbitrariamente como chave 0, e a coordenada y como chave 1.

k-d Tree

- Para cada nível, o discriminador alterna entre x e y .
Portanto, um nodo n do nível 0 (raiz) teria em sua subárvore esquerda apenas nodos cujos valores de x são menores que n_x e em sua subárvore direita nodos cujos valores de x são maiores que n_x . Um nodo m do nível 1 teria em sua subárvore esquerda nodos cujos valores de y são menores que m_y e em sua subárvore direita nodos cujos valores de y são maiores que m_y .

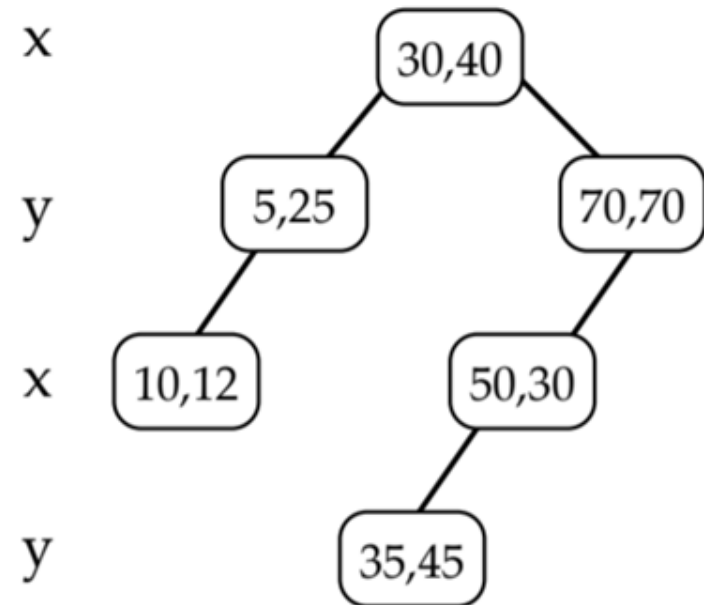
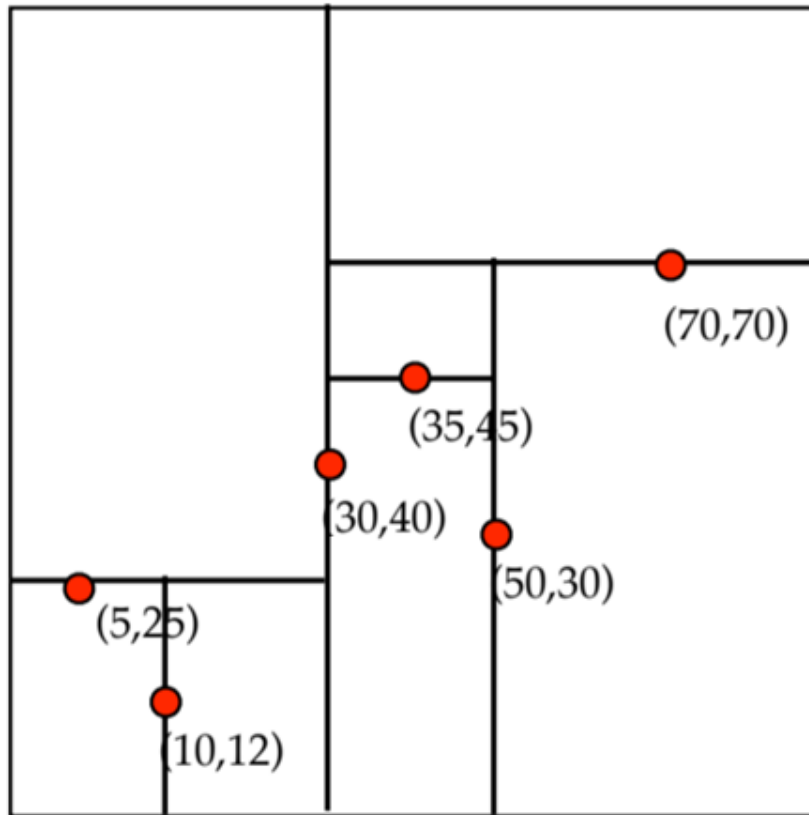
k-d Tree

- Cada nível tem uma "dimensão de corte".
- Percorra as dimensões enquanto caminha pela árvore.
- Cada nodo contém um ponto $P = (x, y)$.
- Para encontrar (x', y') , compara-se as coordenadas da dimensão de corte.
 - se a dimensão de corte é x , a pergunta é: $x' < x$?



k-d Tree – exemplo

- Inserir: (30,40), (5,25), (10,12), (70,70), (50,30), (35,45).



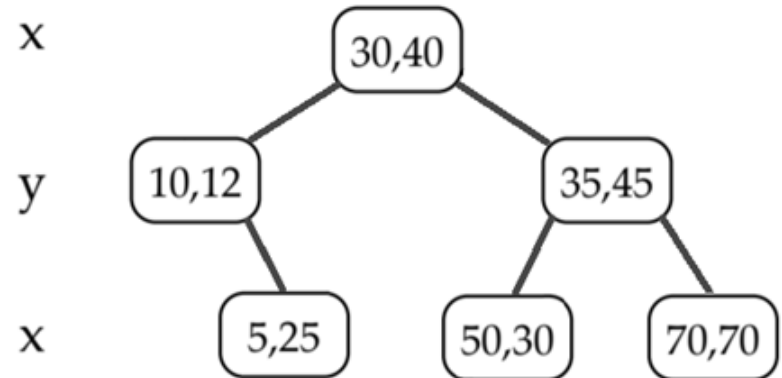
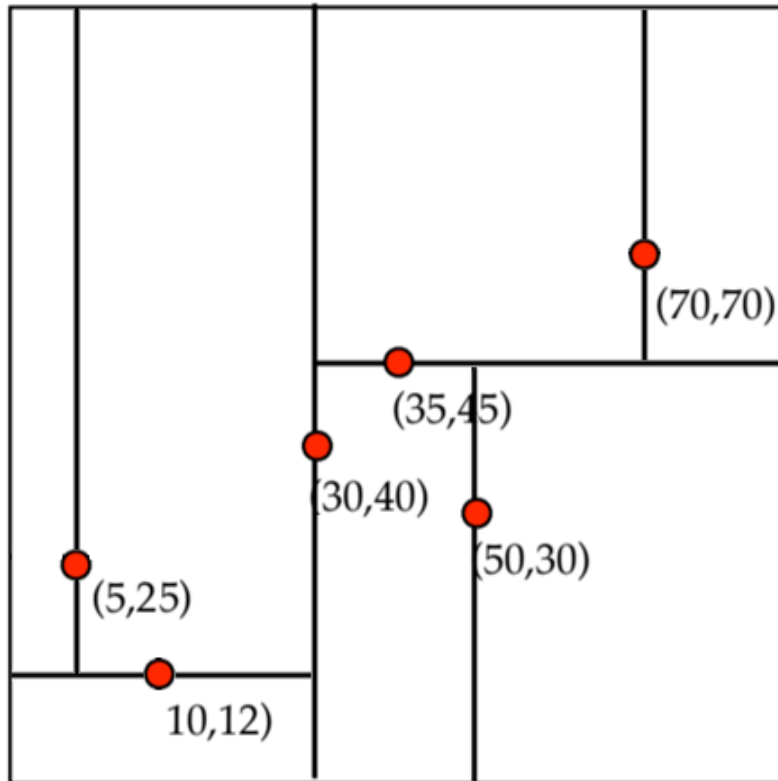
k-d Tree – insere

```
void insereR(kdTree **raiz, int ponto[K], int n)
{
    int d;
    if (*raiz==NULL)
        *raiz=Cria_no(ponto);
    else
    {
        d=n%K;

        if (ponto[d] < (*raiz)->ponto[d])
            insereR(&(*raiz)->esq, ponto, n+1);
        else
            insereR(&(*raiz)->dir, ponto, n+1);
    }
}
```

k-d Tree – insere balanceado

- Inserir: (30,40), (5,25), (10,12), (70,70), (50,30), (35,45).



k-d Tree

- Referência Bibliográfica:
 - Bentley, Jon louis. Multidimensional Binary Search Trees Used for Associative Searching. ACM Student Award, 1975.