

Estruturas de dados avançadas

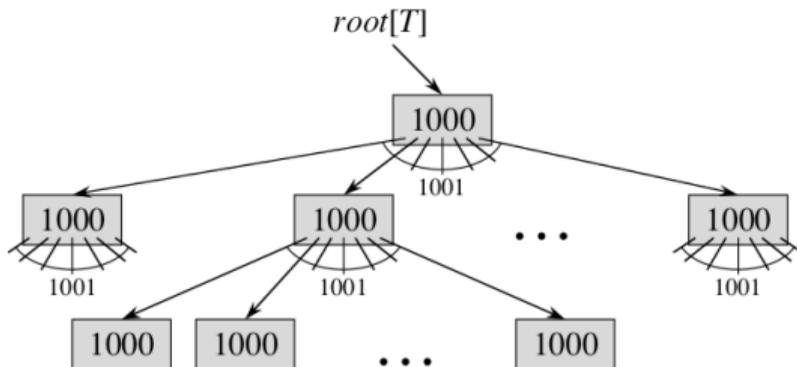
Alessandro Bezerra da Silva

Janeiro de 2022

Vamos falar (bem por cima) de:

- Árvores B
- Árvores *splay*
- Heaps Binomiais
- Heaps de Fibonacci
- Tries

Árvores B



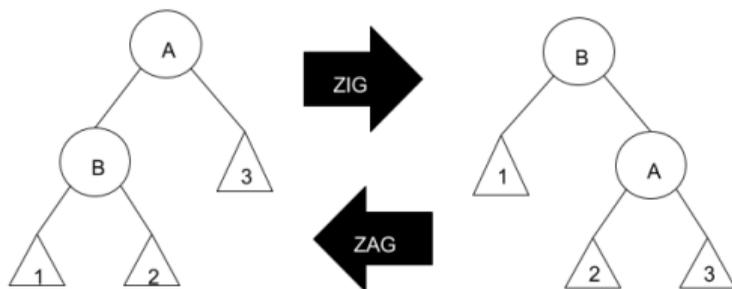
1 node,
1000 keys

1001 nodes,
1,001,000 keys

1,002,001 nodes,
1,002,001,000 keys

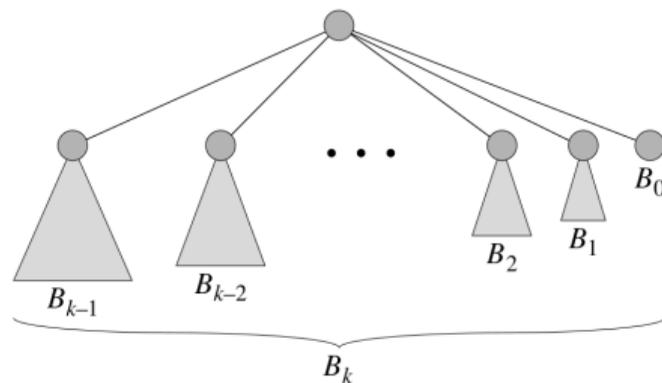
- Desenvolvida para lidar com grandes volumes de dados, cujo conteúdo não cabe inteiramente na memória RAM
- Cada nó da árvore está gravado em disco
- Diferente de ABBs, geralmente possui fator de ramificação grande, tipicamente na ordem de 10^3
- Altura da árvore é pequena, o que implica em poucos acessos ao disco por operação

Árvores *Splay*



- São ABBs, com a propriedade adicional de usar rotações para colocar o elemento inserido ou acessado por último na raiz da árvore
- Elementos mais requisitados são trazidos à raiz, usando-se rotações ZIG/ZAG sucessivas (operação *splay*)
- Tempo amortizado para operações usuais (pesquisa, inserção e remoção) é $\mathcal{O}(\lg n)$

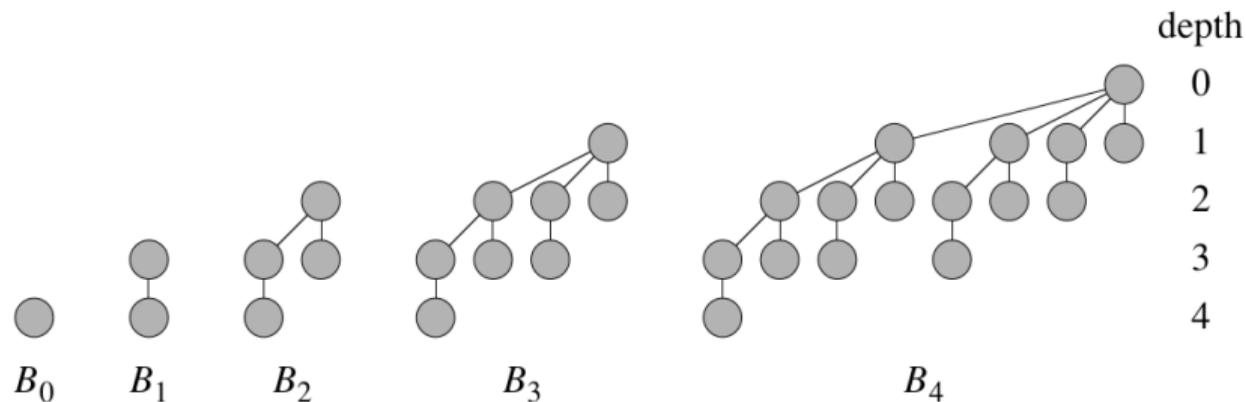
Árvores Binomiais



Estrutura de dados definida recursivamente:

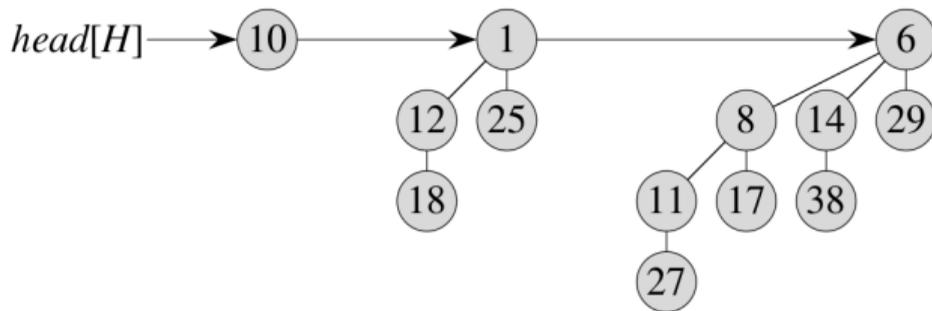
- Uma árvore binomial de ordem 0 (B_0) é um único nó
- Uma árvore binomial de ordem k (B_k) possui um nó raiz, e seus filhos são raízes de árvores binomiais de ordens $k - 1$, $k - 2$, \dots , 1 , 0 (nesta ordem, da esquerda para a direita)

Árvores Binomiais



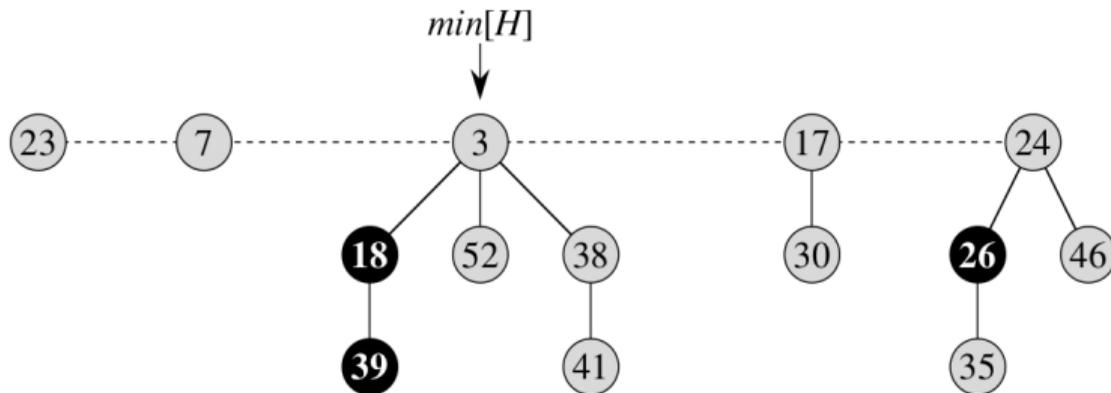
A árvore binomial B_k possui 2^k nós no total, e $\binom{k}{i}$ nós na profundidade i . Por isso a estrutura leva esse nome.

Heaps Binomiais



- Consiste num conjunto de árvores binomiais, representado por uma lista ligada das raízes das árvores binomiais (*lista de raízes*) em ordem crescente
- Possui no máximo uma árvore de cada ordem
- Todas as operações, incluindo a união entre dois heaps binomiais, são executadas em tempo $\mathcal{O}(\lg n)$

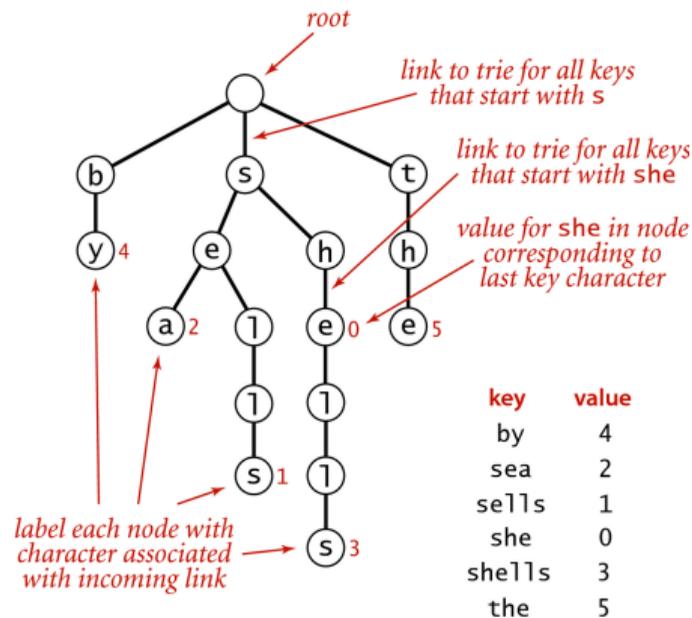
Heaps de Fibonacci



- Estrutura projetada para ter um bom tempo de execução amortizado (oper. s/ remoção de itens = $\Theta(1)$; oper. c/ remoção = $\mathcal{O}(\lg n)$)
- Consiste num conjunto de heaps, dispostos em qualquer ordem e representados por uma lista duplamente ligada com início em $\min[H]$
- Interesse predominantemente teórico (fatores constantes altos)
- Melhora o tempo de execução de algoritmos importantes, por exemplo: algoritmo de Dijkstra

Tries

- Também conhecida como *árvore de prefixos*
- Usada para implementar tabela de símbolos de strings
- Chaves são codificadas nos caminhos que começam na raiz



Anatomy of a trie

O TCC completo estará disponível (em breve) no seguinte link:

linux.ime.usp.br/~alessandrobs/mac0499