MAC121 - Algoritmos e Estruturas de Dados I

Universidade de São Paulo

Segundo Semestre de 2020

Treaps

Árvores de busca binária

Vimos que ABBs são estruturas simples que podemos usar para armazenar tabelas de símbolos.

As operações de busca, inserção e remoção podem ser implementadas em tempo proporcional à altura da ABB.

No pior caso, a altura de uma árvore com n elementos pode ser $n - 1$.

Vimos que, se os elementos forem inseridos aleatoriamente na árvore, a altura esperada será $O(\log n)$.

ABBs balanceadas

Na disciplina MAC0323 – Algoritmos e Estruturas de Dados II vamos aprender formas de garantir que a ABB tenha altura máxima $O(\log n)$.

- Árvores AVL
- Árvores Rubro-negras

Vamos apresentar uma ideia simples que garante com alta probabilidade que a árvore terá altura $O(\log n)$: a treap.

Treap

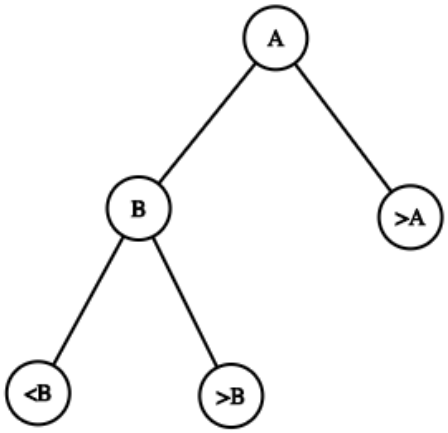
Uma treap (ou ABB cartesiana aleatória) é uma ABB em que cada elemento recebe ao ser inserido um valor aleatório para sua prioridade.

Como em qualquer ABB, os nós satisfazem a propriedade de ordenação dos elementos.

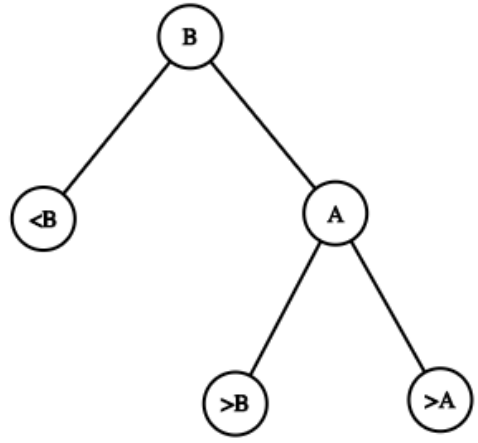
Além disso, as prioridades devem satisfazer a propriedade de um max-heap, definindo a estrutura da árvore.

Treap

Para implementar esta ideia precisamos de duas funções sobre árvores:



rodaDir
$\longrightarrow$
$\longleftarrow$
rodaEsq




```
no * rodaDir (no * raiz);
```

```
no * rodaEsq (no * raiz);
```

Inserção em treaps

O que muda na inserção?

```
se a árvore é vazia, devolve o nó
senão se x < raiz->info
  insere na subárvore esquerda
  se prior(esq) > prior(raiz)
    roda a árvore para a direita
senão
  insere na subárvore direita
  se prior(dir) > prior(raiz)
    roda a árvore para a esquerda
```

Treap

E na remoção, o que muda?

As prioridades podem ser usadas para escolher qual elemento remover (quando o nó tem dois filhos).

Acabou