

Árvore Rubro-Negra

José Félix e Gabriela Pontes

github.com

Motivação

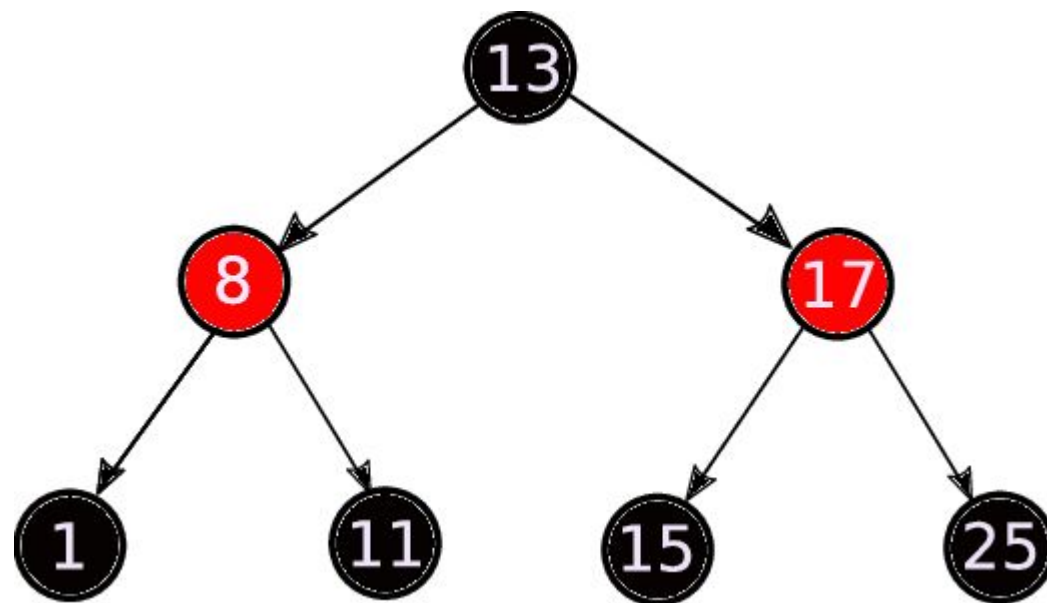
- Bibliotecas, bancos de dados, compiladores
- Inserção e remoção garantidamente em tempo logarítmico $O(\log n)$;
- Implementação mais simples;

Árvore Rubro-Negra

- É um tipo de árvore binária balanceada que utiliza um esquema de coloração dos nós para manter o balanceamento da árvore;
- Originalmente criada em 1972 por Rudolf Bayer chamada de árvores binárias simétricas, e posteriormente em um trabalho de Leonidas J. Guibas e Robert Sedgwick em 1978, adquiriu o nome atual;
- Possui esse nome pois cada em nó dela existe um atributo de cor (além dos ponteiros para os filhos), sendo essas cores o vermelho ou o preto;

Árvore Rubro-Negra

- Possui algumas propriedades que devem ser seguidas:
 - . Todo nó é vermelho ou preto;
 - . A raíz é sempre preta;
 - . Todo nó folha(NULL) é preto;
 - . Não existem nós vermelhos consecutivos;
 - . Para cada nó, todos os caminhos desse nó para os nós folhas descendentes contém o mesmo número de nós pretos.



Árvore Rubro-Negra Caída para a Esquerda

- Em 2008, Robert Sedgwick, desenvolveu a árvore rubro-negra caída para a esquerda, que é uma variante da árvore rubro negra;
- Garante a mesma complexidade de operações, mas possui uma implementação ainda mais simples na inserção e remoção de nós;
- Possui uma propriedade extra: se um nó é vermelho, então ele é o filho esquerdo do seu pai;

Definições

- Coloque aqui as principais definições
- Exemplo que vimos em sala sobre Árvores:
 - Nó, caminho, altura
- Exemplo que vimos em sala sobre Dijkstra:
 - Peso nas arestas

Tipo Abstrato de Dados

```
ArvRB* create_ArvRB();  
void free_ArvRB(ArvRB* raiz);  
ArvRB* insert_ArvRB(ArvRB* raiz, int valor);  
ArvRB* remove_ArvRB(ArvRB* raiz, int valor);  
int isEmpty_ArvRB(ArvRB* raiz);  
int search_ArvRB(ArvRB* raiz);  
void preOrder_ArvRB(ArvRB* raiz);  
void inOrder_ArvRB(ArvRB* raiz);  
void posOrder_ArvRB(ArvRB* raiz);
```

Struct

```
typedef struct node
{
    int data;
    struct node *left;
    struct node *right;
    int color;
} ArvRB;
```


Código - Inserção

Insere valor: 5

5

Insere valor: 30

5

30

Rotaciona à esquerda
em "5"



5

30

Insere valor: 20

5

30

20

Rotaciona à esquerda
em "5"



5

20

30

Rotaciona à direita
em "30"



20

5

30

5

20

30

Troca cor em "20"
Raiz fica preta



Figura 12.82

Insere valor: 10

5

20

30

Rotaciona à esquerda
em "5"



5

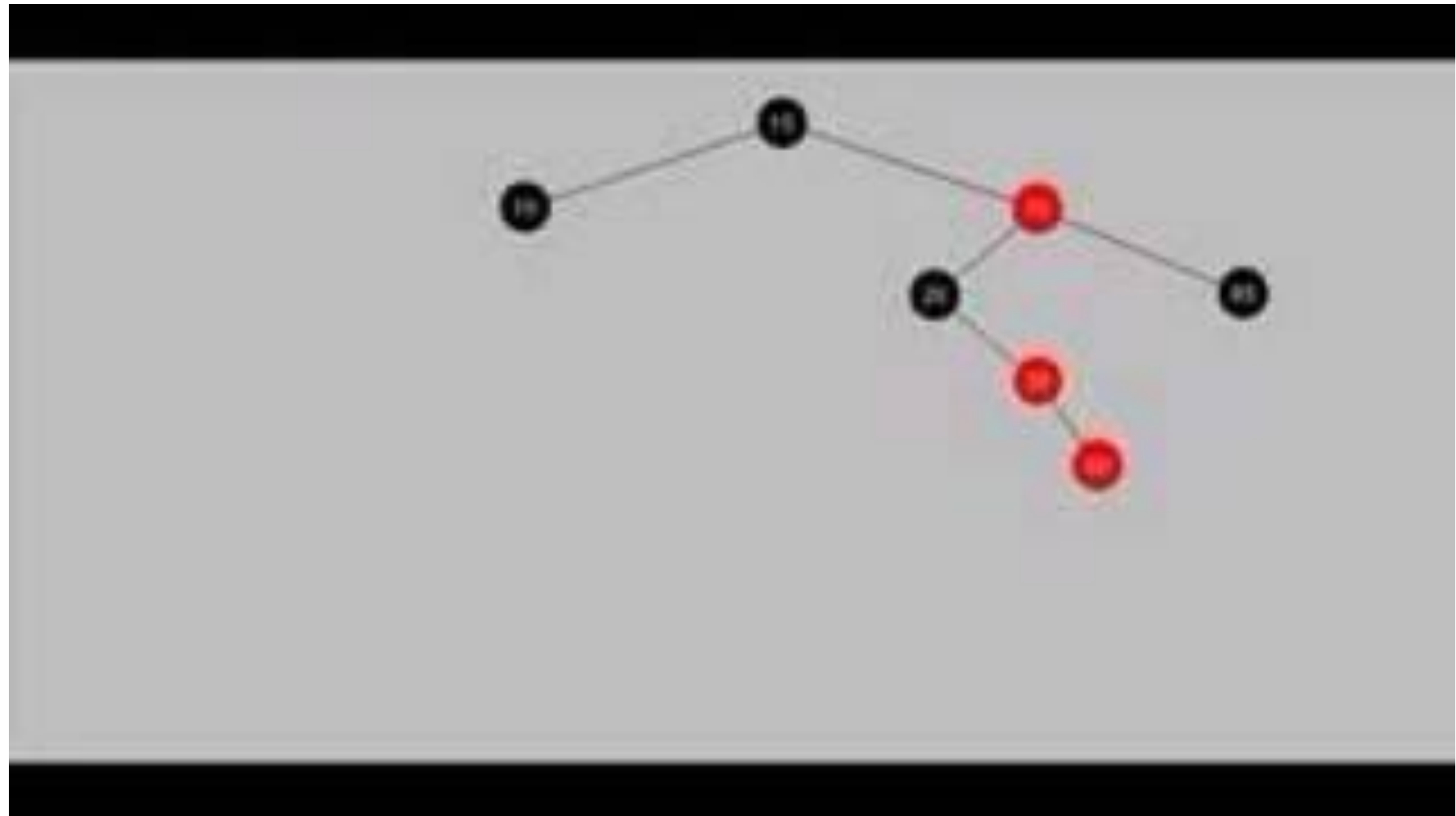
10

20

30

Figura 12.83

Animação - Inserção de nós em uma Árvore Rubro-Negra



De volta à Motivação...

- Por que a estrutura/algoritmo resolveu o problema descrito na motivação?!