A árvore rubro-negra

Agora a gente tem a lógica da árvore 2-3 funcionando em uma árvore binária de busca.

Mas isso ainda não simplificou as coisas — (*porque ainda existem casos demais*)

Para conseguir isso, a gente vai permitir que os elementos pretos também tenham um filho esquerdo vermelho.

Porque daí, a inserção à esquerda de um elemento preto sempre pode ser resolvida criando o filho vermelho.

Certo.

Isso simplifica as coisas.

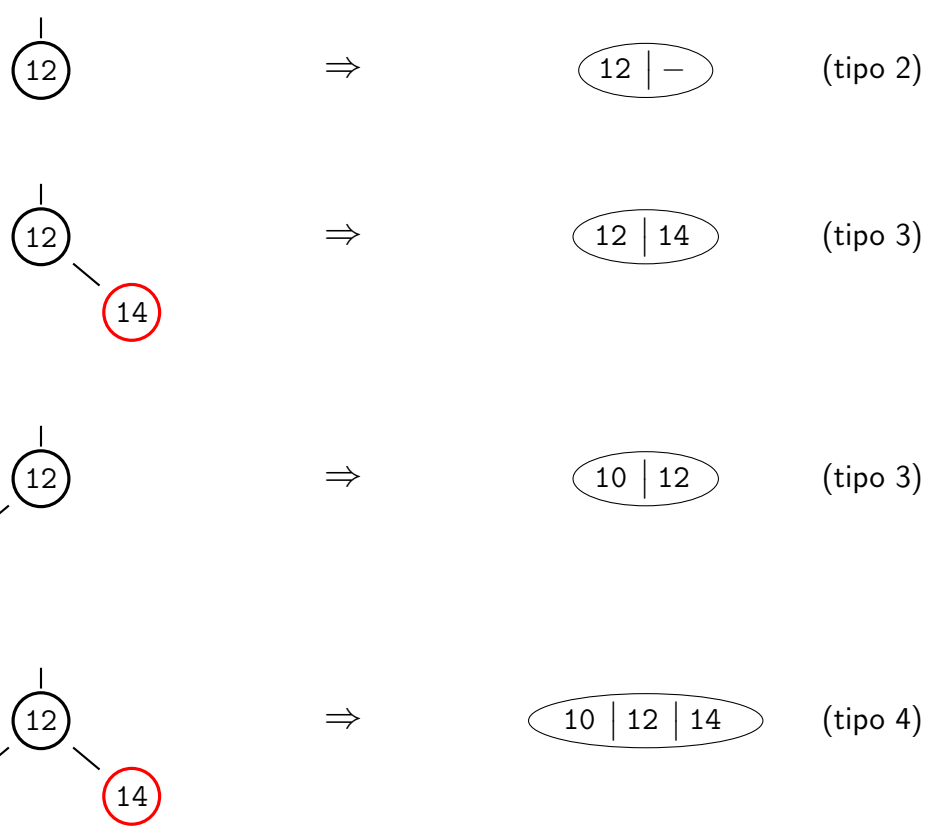
Mas a gente perde a interpretação da árvore 2-3.

E ao perder a interpretação, a gente perde a garantia da altura $O(\log n)$.

Só que, é bem fácil recuperar essa garantia.

Quer dizer, basta encontrar outra interpretação.

E a interpretação é que agora a nossa árvore é uma árvore 2-3-4 desmontada.



Abaixo nós temos um exemplo com a estrutura da árvore 2-3-4 em destaque

(. . .)

Olhando para esse exemplo, a gente vê que

- Os elementos vermelhos são sempre o filho de um elemento preto
- Os elementos pretos formam uma árvore completa — (*porque a árvore 2-3-4 é uma árvore completa*)

Como a subárvore preta é completa, ela tem altura $O(\log n)$.

Daí a gente nota que os elementos vermelhos no máximo dobram a altura da árvore — (*porque?*)

E isso já nos dá a garantia de que a nossa árvore tem altura $O(\log n)$.

Essa árvore é conhecida como a *árvore rubro-negra*.