Fazendo mais anotações

E agora?

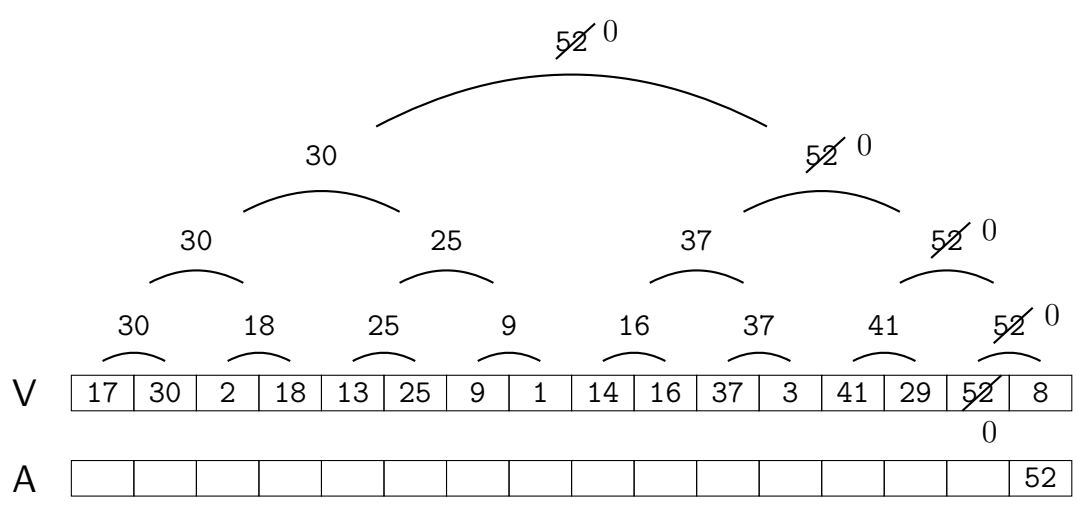
Bom, agora a gente usa a mesma ideia outra vez.

Quer dizer, depois que a gente descobre quem é o maior, a gente pode esquecer dele.

Contanto que ele fique anotado em algum lugar.

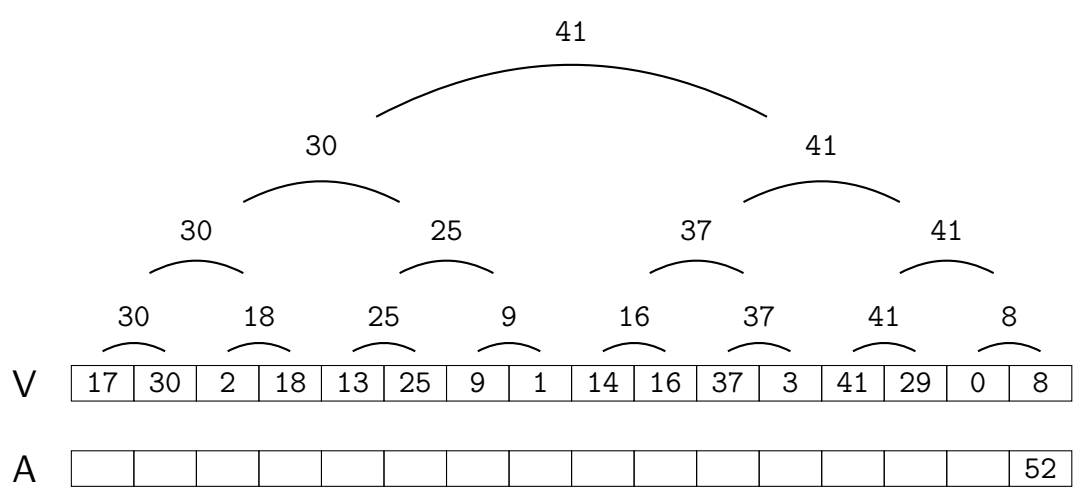
Então agora, a gente vai ter uma lista auxiliar — (*para guardar as anotações*)

E para esquecer o maior, a gente muda o seu valor para 0 — (*em todos os lugares onde ele aparece*)



Daí, como o zero é menor do que todo mundo, ele perde todas as comparações

E a coisa é reconstruída assim

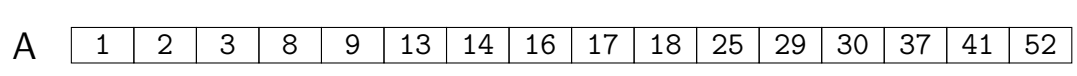


Pronto! agora nós temos a mesma coisa outra vez.

Quer dizer, o maior elemento está no topo — (*dentre aqueles que sobraram*)

E nós podemos fazer a mesma coisa para colocá-lo na lista auxiliar, e restaurar a situação no mesmo formato.

Daí que, fazendo isso repetidamente, a gente obtém a lista completamente ordenada.



Legal.

Análise

Mas, quanto tempo leva isso?

Bom, a coisa funciona em duas etapas

1. a construção da estrutura de árvore
2. a construção da lista auxiliar ordenada

E a gente pode analisar o tempo de cada etapa em separado.

A gente estima o tempo da primeira etapa contando o número de comparações que são realizadas ali

- primeiro, a gente compara os pares de elementos consecutivos

$$\frac{n}{2} \quad \text{comparações}$$

- depois, a gente compara os ganhadores dessas comparações

$$\frac{n}{4} \quad \text{comparações}$$

- depois, a gente compara os ganhadores dessas comparações

$$\frac{n}{8} \quad \text{comparações}$$

- e por aí vai ...

Daí, a gente soma tudo para obter

$$\frac{n}{2} \quad + \quad \frac{n}{4} \quad + \quad \frac{n}{8} \quad + \quad \ldots \quad + \quad 2 \quad + \quad 1 \quad = \quad O(n)$$

Certo.

Agora a gente considera a segunda etapa.

Na segunda etapa, a gente

- copia o maior elemento para a lista auxiliar
- esquece o maior elemento — (*substituindo o seu valor por 0*)
- e refaz as comparações ao longo desse caminho

Daí, a gente nota que o caminho do topo até embaixo tem tamanho $\log_2 n$ — (*porque?*)

Logo, a coisa toda leva tempo

$$O(1) \quad + \quad O(\log n) \quad + \quad O(\log n) \quad = \quad O(\log n)$$

Mas, isso precisa ser feito uma vez para cada elemento.

Logo, a coisa toda leva tempo

$$O(n \log n)$$

Finalmente, é preciso somar os tempos das duas etapas

$$O(n) \quad + \quad O(n \log n) \quad = \quad O(n \log n)$$

Legal, não é?