

Sim, mas nós ainda não conseguimos um algoritmo mais rápido do que $O(n^2)$.

Então agora, nós vamos ver a nossa esperteza final.

Quer dizer, agora o nosso problema é o seguinte

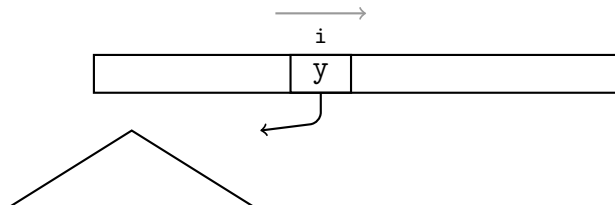
→ *Contar, para cada elemento da lista J*

quantos elementos à sua esquerda são menores do que ele

quantos elementos à sua direita são maiores do que ele

E para fazer isso de maneira eficiente, nós vamos utilizar árvores binárias balanceadas.

A ideia consiste em percorrer a lista J, e inserir cada um deles na árvore



Daí, como a árvore está balanceada, cada inserção leva tempo $O(\log n)$.

E daí, a esperteza consiste em contar a quantidade de elementos menores do que y na árvore durante o próprio procedimento de inserção.

Para isso, basta que cada nós da árvore armazene a quantidade de elementos nas suas subárvores esquerda e direita.

Daí, o número de elementos menores do que y na árvore é o número de elementos menores do que y à esquerda de y na lista J.

E com esse número, nós conseguimos determinar o número de elementos maiores do que y à direita de y na lista J.

Portanto, em tempo $O(\log n)$ nós conseguimos determinar o número de triplas engraçadas pivoteadas por cada elemento.

E isso nos dá um algoritmo de tempo $O(n \log n)$ para o nosso problema.

Legal, não é?