

03

Como ordenar ao pivotar

Transcrição

Conseguimos encontrar a posição exata que o pivô deveria ficar. Mas faltou separar os elementos menores para a esquerda do *array*. Nós vimos como fazer isto... Quando estamos dentro do laço e encontramos alguém menor, qual é o próximo passo? Devemos trocar o elemento que está na posição `analisando` para a posição `menoresEncontrados`.

```
troca(notas, analisando, menoresEncontrados);
```

O laço ficará assim:

```
Nota pivo = notas[termino - 1];
for(int analisando = 0; analisando < termino - 1; analisando++) {
    Nota atual = notas[analisando];
    if(atual.getValor() <= pivo.getValor()) {
        troca(notas, analisando, menoresEncontrados);
        menoresEncontrados++;
    }
}
troca(notas, analisando, menoresEncontrados);
```

Se temos três notas menores até agora, então as posição 0, 1 e 2 são para elementos menores. Mas a casinha 3 será maior. Então, trocarei o elemento que estou analisando de posição. Se temos 17 elementos que são menores até agora, da posição 0 até 16, todos os itens serão menores. O que farei é colocar o pivô na casinha 17. Em seguida, somarei +1 na variável `menoresEncontrados`.

Testaremos novamente e veremos o que acontecerá. Clicamos em *Run As*, depois em *Java Application*, e o resultado será:

andre 4.0

jonas 3.0

juliana 6.7

mariana 5.0

guilherme 7.0

lucia 9.3

paulo 9.0

carlos 8.5

ana 10.0

Os quatro primeiros elementos da lista (André, Jonas, Juliana e Mariana) têm notas menores que o pivô (Guilherme). Todos que eram menores foram particionado para a direita e todos que eram maiores foram para a direita. Agora, nós efetivamente quebramos o *array* no pivô. Também queremos ser capazes de informar em qual posição o Guilherme ficou. Então iremos retornar no fim `menoresEncontrados`.

```
Nota pivo = notas[termino - 1];
for(int analisando = 0; analisando < termino - 1: analisando++) {
    Nota atual = notas[analisando];
    if(atual.getValor() <= pivo.getValor()) {
        troca(notas, analisando, menoresEncontrados);
        menoresEncontrados++;
    }
}
troca(notas, analisando, menoresEncontrados);
return menoresEncontrados;
```

O código irá nos informar "o Guilherme foi para essa posição".

O nosso método irá devolver um `int` e também teremos a nova posição:

```
int novaPosicao = quebraNoPivo(notas, 0, notas.length);

for(int atual = 0; atual < notas.length; atual++) {
    Nota nota = notas[atual];
    System.out.println(nota.getAluno() + " " + nota.getValor());
}
```

Se quisermos, podemos imprimir: "O pivô foi parar em " + `novaPosição`:

```
int novaPosicao = quebraNoPivo(notas, 0, notas.length);

System.out.println("O pivô foi parar em " + novaPosição);
```

Ao rodarmos novamente, o programa informará que o pivô foi colocado na posição 4.

O pivô foi parar em 4

andre 4.0

jonas 3.0

juliana 6.7

mariana 5.0

guilherme 7.0

lucia 9.3

paulo 9.0

carlos 8.5

ana 10.0

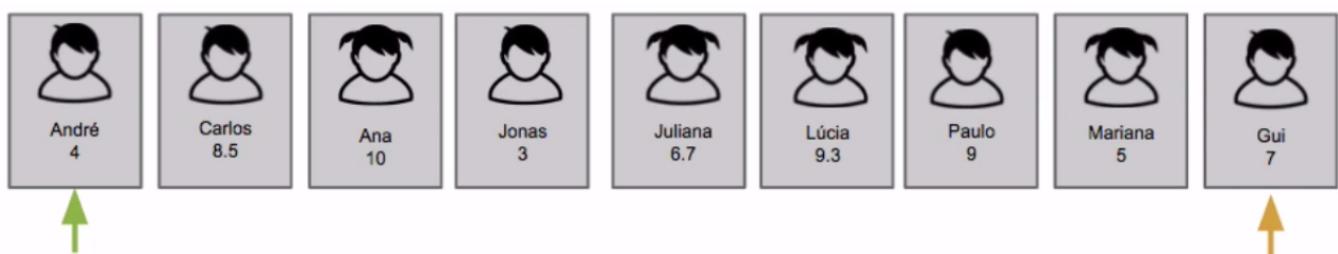
A função `quebraNoPivo()` funciona. Ela encontra a posição adequada para o pivô e depois quebra o `array` em duas partes: os menores e os maiores.

Simulando a partição

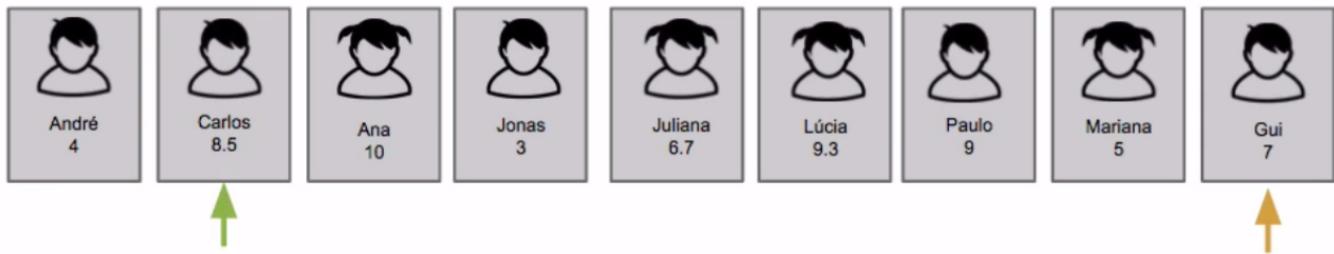
Queremos varrer o nosso `array`, contando os `menoresAtéAgora`, porque assim poderemos encontrar a posição do pivô. Ao descobrirmos a posição adequada, trocaremos o elemento de lugar e ficaremos satisfeitos...

Mas apenas isto, não é o suficiente. Queremos também saber quem foi melhor ou pior e que os elementos menores fiquem posicionados à esquerda. À medida que vamos varrendo e analisando os elementos do `array`, nós queremos de alguma maneira informar: "se o elemento é menor do que o Guilherme, ele deve ficar à esquerda". Analisando cada elemento, identificamos se ele deve ser posicionado na parte esquerda do `array`. Caso contrário, não é preciso movê-lo.

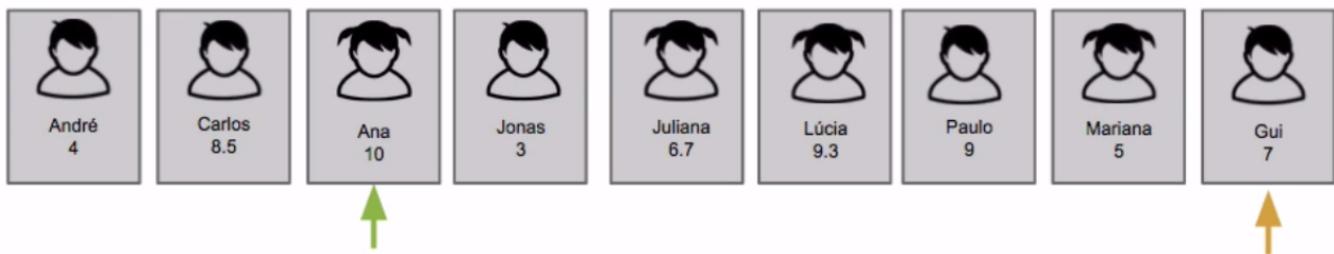
Começaremos a trabalhar com o algoritmo. As variáveis `analisando` e `menoresAtéAgora` serão igual a 0. E o pivô estará na posição 8. O elemento analisado será o André.



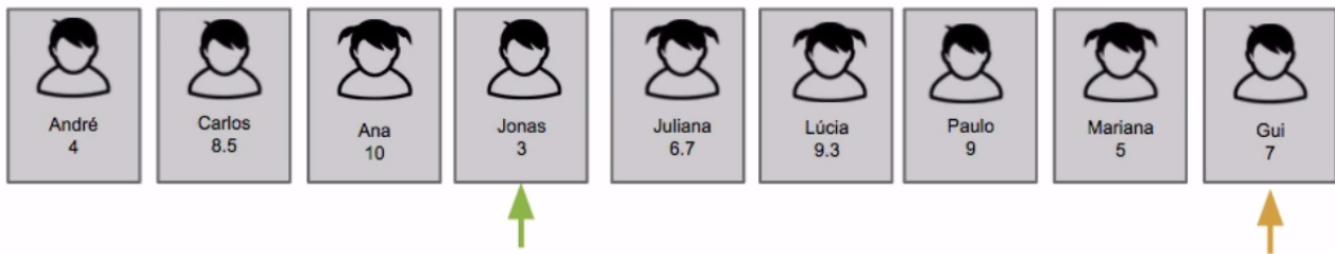
A nota 4 é menor do que 7? Sim. Sendo assim, vou manter o elemento à esquerda, na posição 0. `menoresAtéAgora` e `analisando` serão igual a 1. Por enquanto, funciona o algoritmo. Seguiremos para o Carlos.



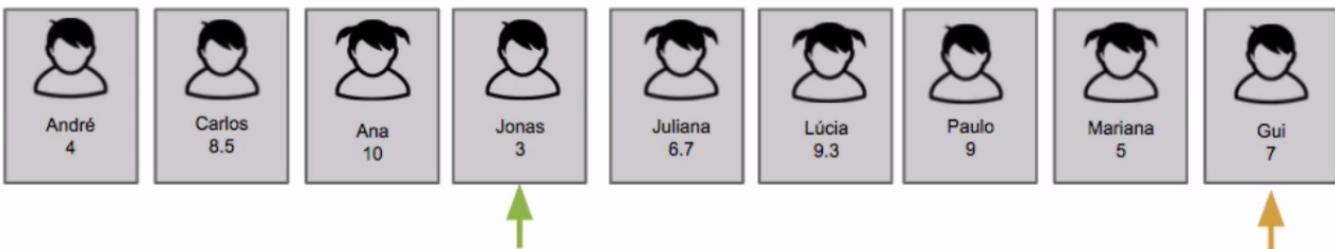
A nota 8.5 é menor ou maior do que 7? É maior. Não faremos alterações e seguiremos para o próximo. `analizando` será igual a 2.



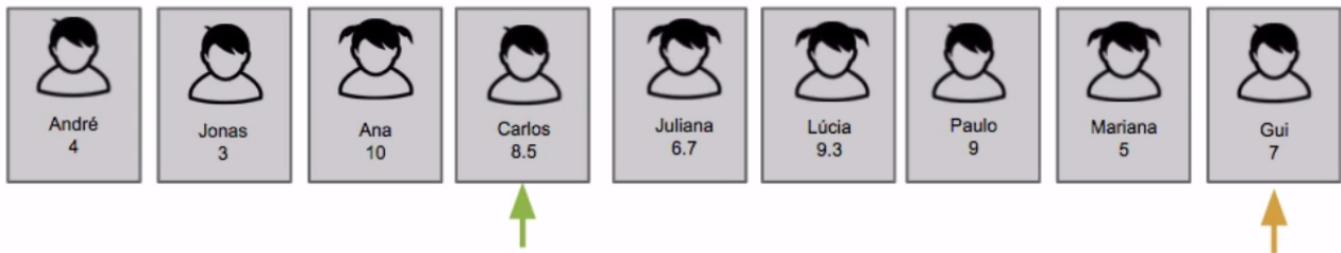
A Ana tirou nota maior ou menor do que o Guilherme? Ela tirou 10, uma nota explicitamente maior. Vamos para o elemento seguinte. Somaremos +1 na variável `analizando`.



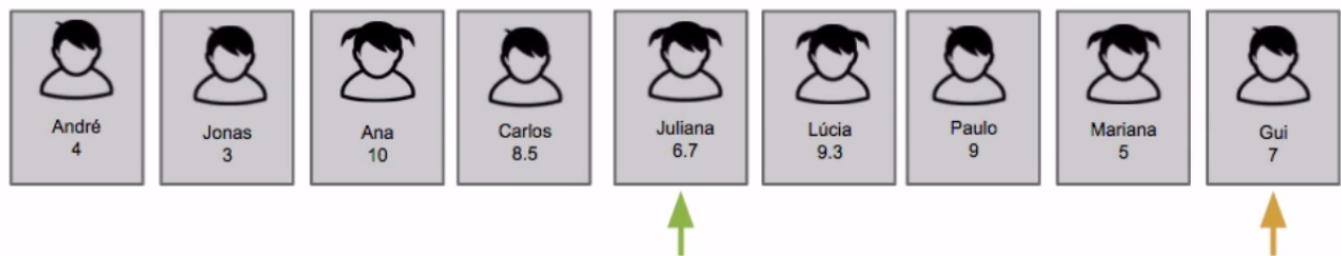
Agora aumenta a dificuldade... O Jonas é menor do que o Guilherme? Sim. Se ele é menor, o que precisamos fazer? Colocá-lo mais à esquerda no array. Iremos trocá-lo de posição. Com qual elemento? Não precisamos tirar o André da posição 0, porque ele também é menor do que o pivô. E se trocar os dois elementos de posição o André não ficará em um lugar adequado.



Queremos trocar o Jonas de lugar com o primeiro elemento que não seja menor que o Guilherme. Quantos são menores até agora? Apenas **um**. Isto significa que devemos colocar o Jonas na posição do `menoresAtéAgora`. O Carlos que ocupa a posição 1, não é menor que o pivô. Então podemos trocar os elementos de posição.



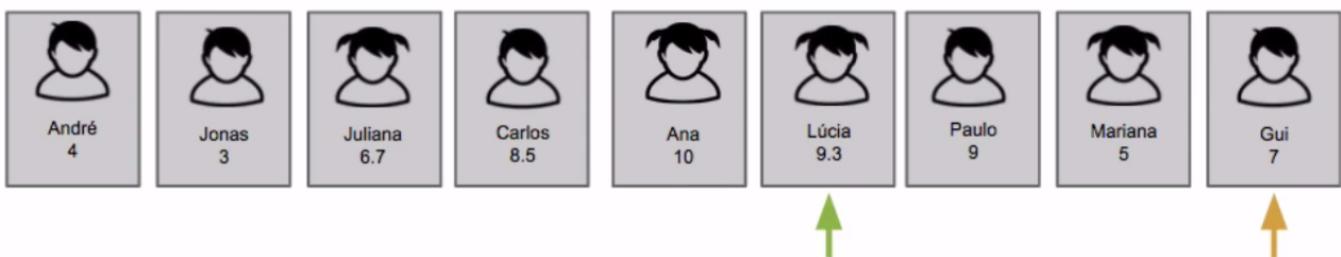
Seguiremos com o processo de sempre. Somaremos +1 nas variáveis `menoresAtéAgora` e `analizando`. Vamos para o próximo.



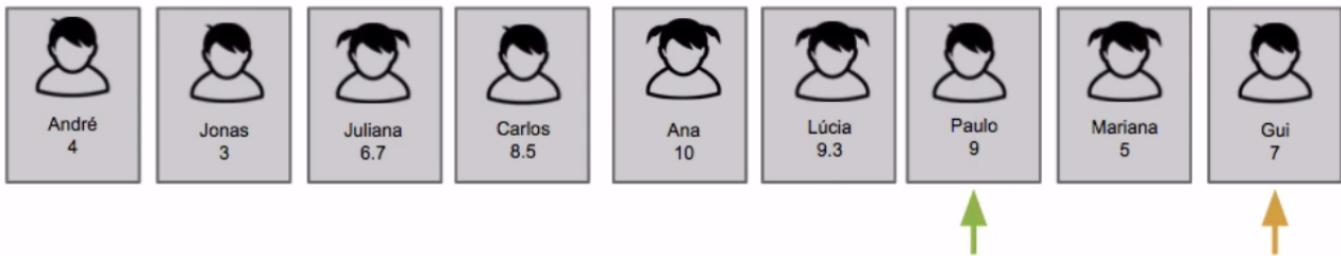
A Juliana é maior ou menor? Menor. Como ela tirou um nota menor, iremos trocá-la de lugar. Mas para qual posição? Não será nas posições 0 e 1, porque iria atrapalhar o processo. A posição correta será a 2, afinal é o número de `menoresAtéAgora`. Iremos trocá-la de lugar com a Ana.



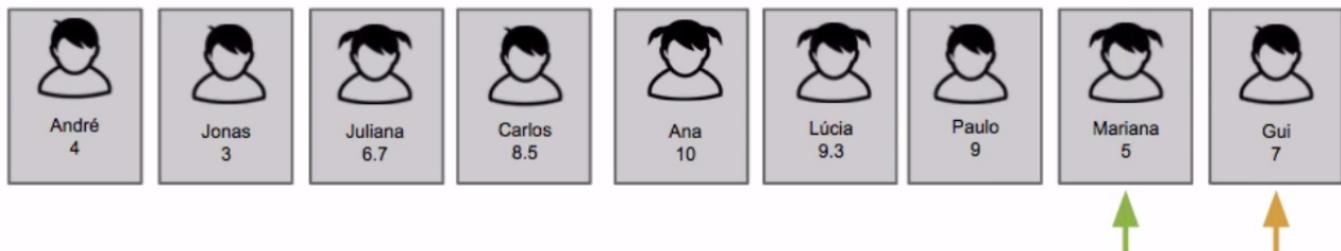
Agora que encontramos outro elemento menor, somaremos + 1 na variável `menoresAtéAgora`. Aumentaremos também o valor de `analizando` que será igual a 5. Vamos para o próximo.



A Lúcia é maior do que o Guilherme? Sim. Então apenas seguiremos para o próximo. `analizando` será igual a 6.



O Paulo é maior? Sim. Então vamos para o próximo. Somaremos +1 no analisando .



E a Mariana, é menor ou maior? Menor. Então vamos ter que trocá-la de posição. Não podemos movê-la para as posições 0, 1 e 2, caso contrário, algum elemento menor ficaria posicionado à direita. Temos que colocá-la na posição 3. Por quê? Porque temos três no menoresAtéAgora . Então iremos trocar a Mariana com que estiver na posição 3, no caso será com o Carlos.