

## Ordenação e baralho: Juntando dois conjuntos ordenados

### Transcrição

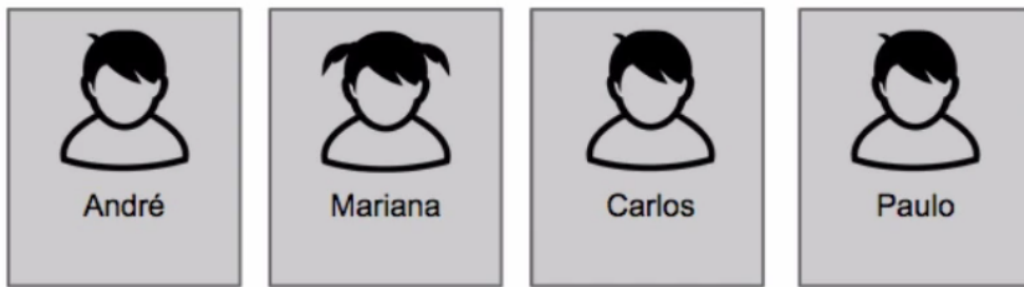
O que acontece quando eu recebo cinco cartas de baralho? Em diversos jogos é interessante que as cartas menores fiquem separadas em uma parte, enquanto as maiores fiquem em outra. Assim, fica mais fácil encontrar os elementos que tenho na mão. Como é costume trabalhar desta maneira, é comum que as pessoas façam esta ordenação manualmente.

É simples ordenar cinco cartas do poker, três cartas do truco. Mas quando ordenamos treze cartas do buraco, já aumenta a dificuldade. Quanto maior for a quantidade de itens para serem ordenados, mais trabalhoso será o processo. Por quê? Porque os algoritmos que as pessoas usam são o *Selection Sort* e *Insertion Sort*, e à medida que o número de elementos cresce, eles começam a ficar muito lentos. Serão muitas operações e trocas, e isto justifica a demora. Nós não queremos passar por isto. Vamos usar um exemplo que tenha mais de 13 cartas? Imagine um jogo de buraco ou de poker, em que no fim da partida todas as cartas estejam na mesa. E então, você pede para que alguém ordene todas as cartas... Ninguém gostaria de fazer isto. Outro exemplo, seria o jogo da vida. No fim da partida, precisamos que alguém reordene as notas do dinheirinho falso. Ninguém quer ser a pessoa que executará a tarefa de ordenação. No fim do jogo de baralho, do jogo da vida, ou organizar as notas de 300 alunos, ninguém quer ficar responsável por ordenar, porque são muitos elementos. É trabalhoso fazer um *Selection Sort* e um *Insertion Sort*, porque teremos que executar muitas operações. Será que não existe outros algoritmos ou processos mais rápidos?

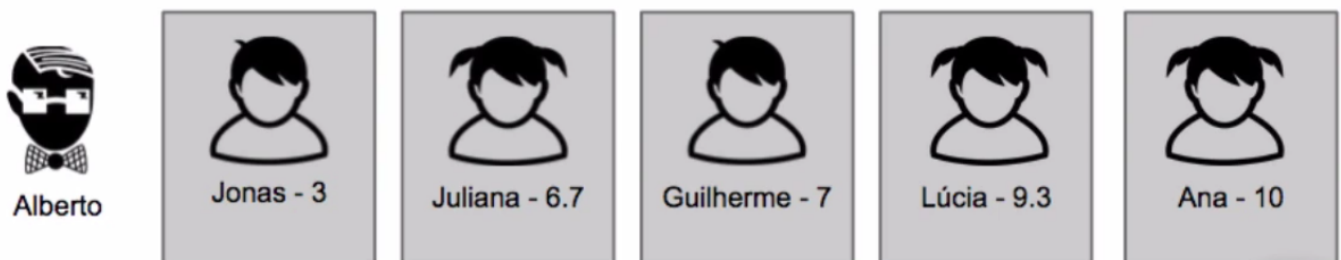
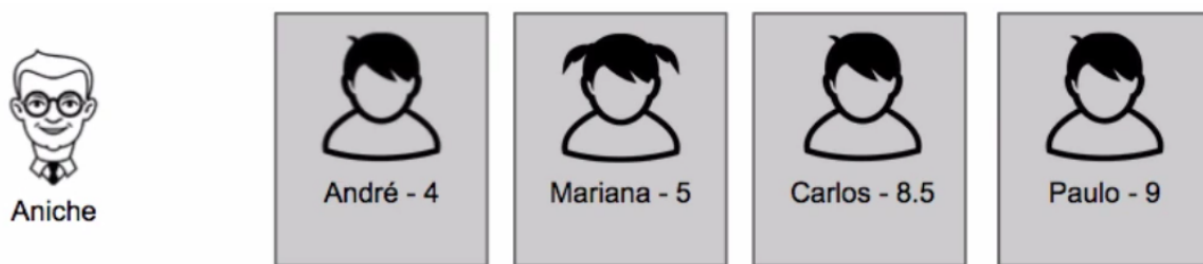
Eu gosto de encontrar formas mais simples, e por isso, vou contar um segredo que costuma funcionar e é utilizado por muitas pessoas... Quando o jogo de buraco acaba e todas as cartas estão na mesa, se alguém decide ordenar os itens, devemos agrupar todos eles e dividi-los entre os participantes. Se temos um monte de cartas - ou de dinheirinho - e tivemos uma partida com quatro jogadores, distribuímos 1/4 do total de elementos para cada participante e todos serão responsáveis pela ordenação da parte recebida. **Ordenar 1/4 do total de cartas é menos trabalhoso do que organizar todos os itens!** Após cada um ordenar a sua parte, reagrupamos as cartas em um único monte que já estará organizado.

O que nós fizemos? Quebramos o nosso problema em quatro pedaços (poderiam ter sido um número maior ou menor) e depois de organizados, juntamos as partes. Podemos fazer o mesmo processo com as provas de alunos. Por exemplo, vamos corrigir as provas do Enem e precisamos ordenar 1 milhão delas. Que trabalhoso executar o *Insertion Sort*! E se encontrássemos uma outra maneira de fazer a ordenação?

Vamos usar a maneira que já conhecemos ao organizarmos as cartas de baralho, o dinheirinho do Jogo da Vida... Usaremos o exemplo das provas dos nove alunos:



Ao invés de ordenar todos os itens, eu divido os alunos entre duas pessoas: o Aniche e o Alberto. Então, cada um deles será responsável pela ordenação considerando as notas dos alunos. Vamos observar como ficam os grupos do Aniche e do Alberto ordenados:



Agora que já temos os dois grupos ordenados, o nosso trabalho será uni-los. A sacada quando trabalhamos com um número grande de elementos, é dividir a tarefa entre as pessoas. No nosso exemplo, o grupo de aluno foi dividido entre o Aniche e o Alberto e cada um teve que organizar a metade.

Então, o problema que quero resolver é: considerando o grupo do Aniche e o grupo do Alberto já ordenados - ou seja, dados os *arrays* ordenados dos dois - o que teremos que fazer é intercalar os elementos das duas sequências para ter um *array* todo organizado.

Então, dado o grupo do Aniche e o do Alberto, o que farei é intercalar todos os objetos para que eles fiquem ordenados em um único *array*. Dado dois *arrays* ordenados, intercale os elementos e monte um *array* ordenado. Se formos capazes de fazer isto, será um enorme avanço, porque basta dividir a ordenação entre as pessoas e depois, unir os elementos ordenados - tarefa que já sabemos fazer.

Vamos tentar unir dois *arrays* de uma maneira mais rápida do que faríamos com outros algoritmos. E assim conseguir ordenar uma quantidade de dados muito maiores. Com o algoritmos que estamos montando, nós seremos capazes. O

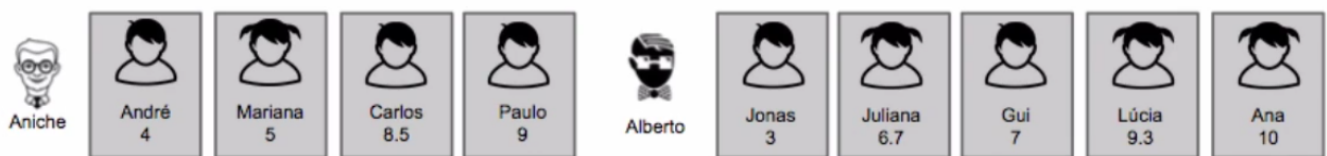
primeiro passo será: considerando que os grupos do Aniche e do Alberto já estão ordenados, como iremos unir estes dois?

## Como juntar ou intercalar duas listas ordenadas

Baseado no que usamos cotidianamente, no jogo de truco, no buraco, no Jogo da Vida, qualquer atividade que exija a distribuição de diversos itens. Podemos distribuir entre os participantes para que todos ajudem a organizar e depois juntamos tudo novamente. Queremos resolver o problema de reagrupar os elementos.



Considerando os alunos do Aniche e do Alberto, que já foram ordenados, como podemos intercalar os nove alunos? Teremos trabalhar com a variável de 9.



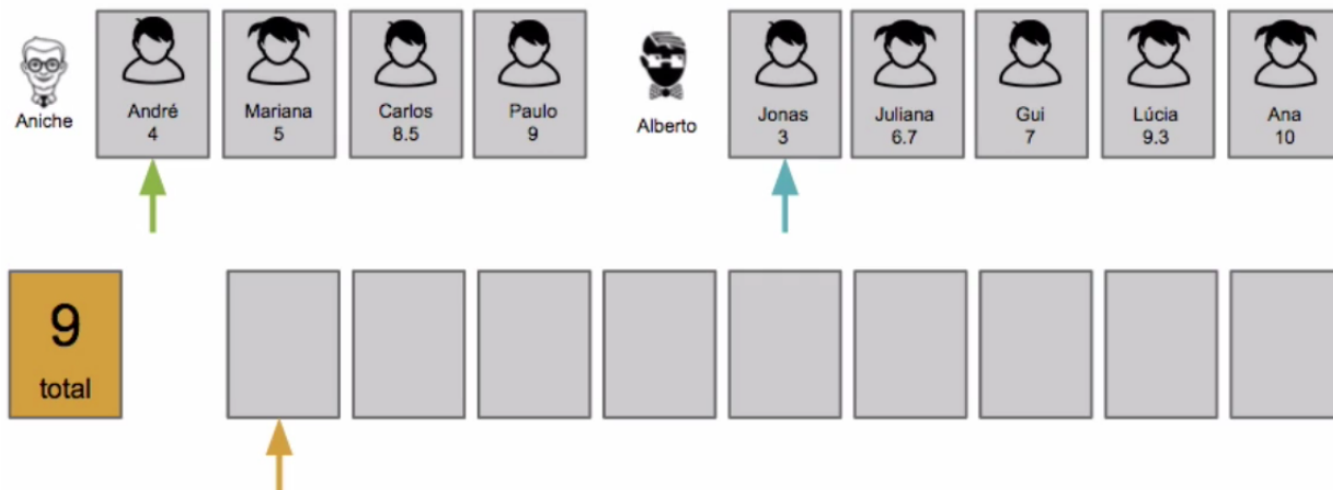
9  
total

Sempre que estamos trabalhando com ordenação, em linguagem de programação que o `array` já nos diz qual o número total de elementos, nós conseguimos extrair o número total. Como queremos generalizar para qualquer linguagem de programação, temos que saber o total de alunos. Que no caso são 9 alunos, iremos reagrupá-los em um único `array` que caiba todos eles. Então, nosso primeiro passo será criar um `array` com um tamanho que caiba os 9 elementos.

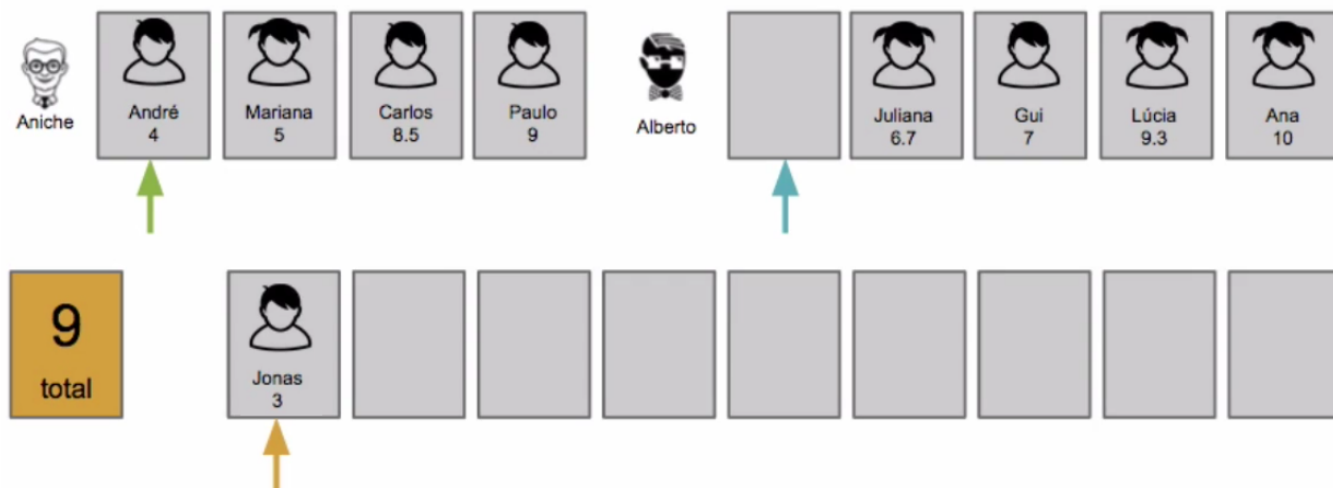


Caso contrário, não teremos como juntar todos os elementos em um único `array`. Agora que temos um lista com tamanho 9, podemos começar.

Iremos iniciar com o primeiro elemento de cada grupo ordenado, porque começar pelo meio não faria sentido. Seguiremos a opção mais simples.



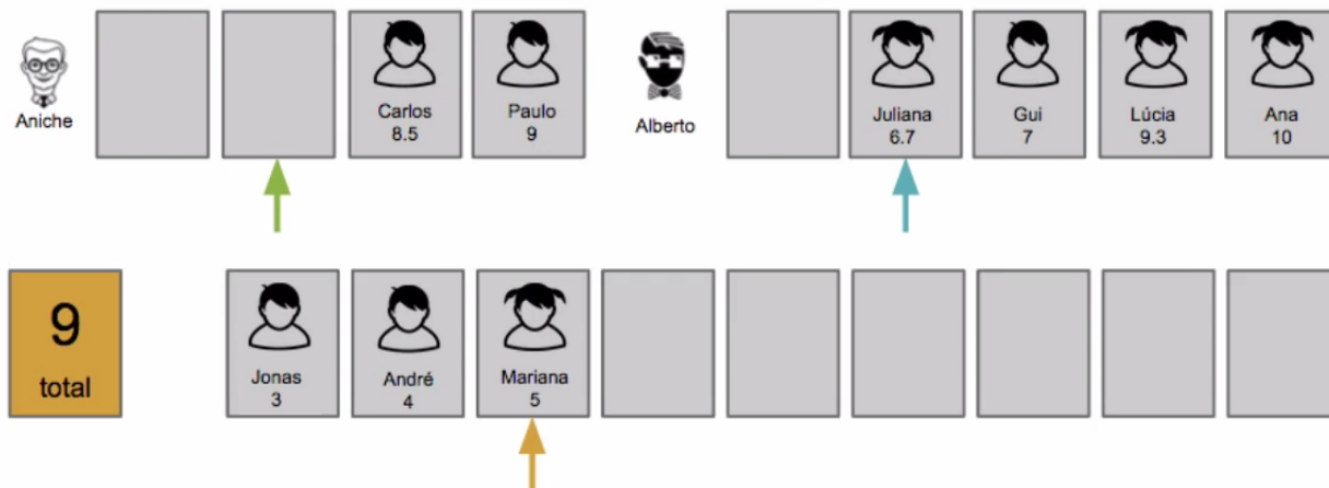
Vamos comparar o André com o Jonas. Um teve uma nota 4 e o outro teve uma nota 3. O Jonas não se saiu bem nas provas! Ele será o menor do nosso novo *array*.



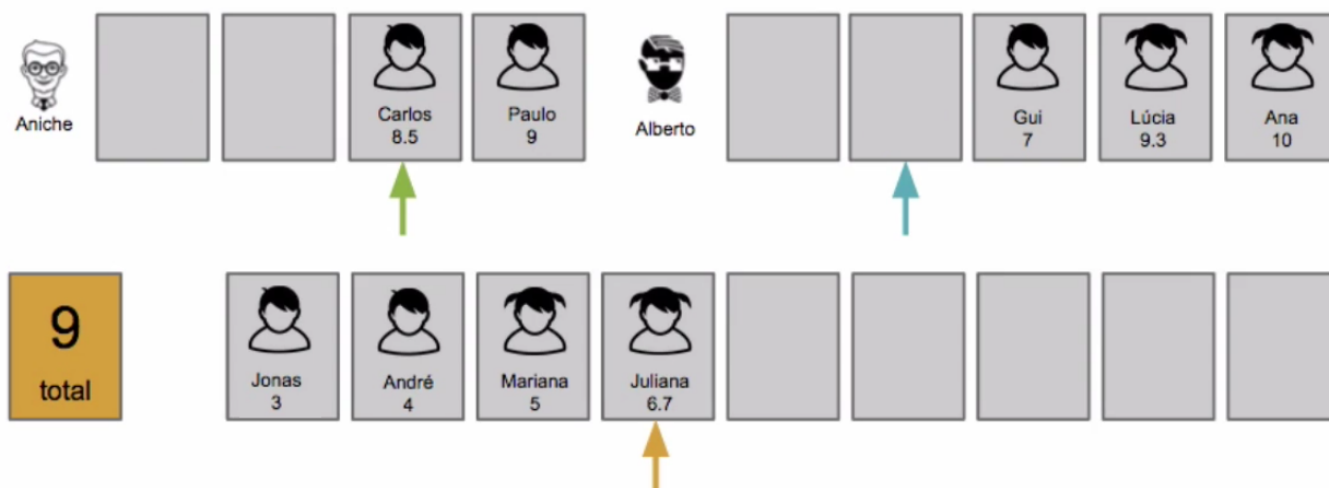
Como já colocamos o Jonas no início da lista, vamos comparar agora o próximo dos grupos do Aniche e do Alberto: o André e a Juliana. Um aluno tirou uma nota 4 enquanto o outro tirou uma nota 6,7. Qual dos dois tirou uma nota menor? O André. Vou descer o André para o *array*.



Seguimos comparando as próximas alunas de cada grupo: Mariana e Juliana. Qual das duas tirou uma nota menor? A Mariana. Vamos desce-la para o *array*.



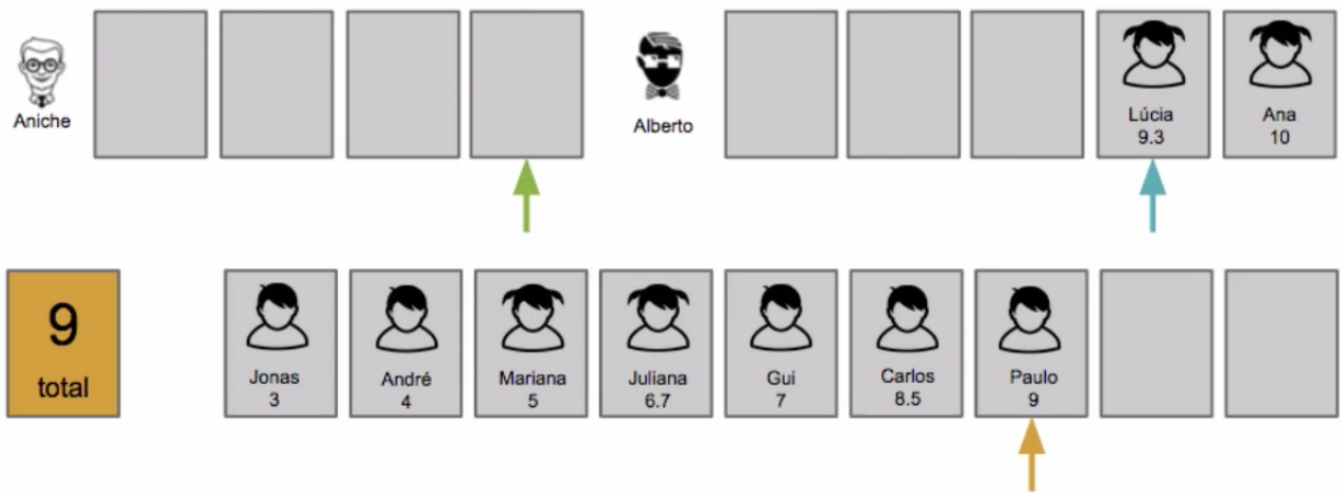
Os próximos alunos a terem as notas comparadas serão: o Carlos e a Juliana. Quem se saiu pior na prova? A Juliana. Vamos desce-la para o novo *array*.



Iremos comparar as notas do Guilherme e do Carlos. O Gui tirou uma nota 7, enquanto o Carlos tirou 8,5. Quem teve a menor nota? O Guilherme. Então, vamos desce-lo para o *array*.

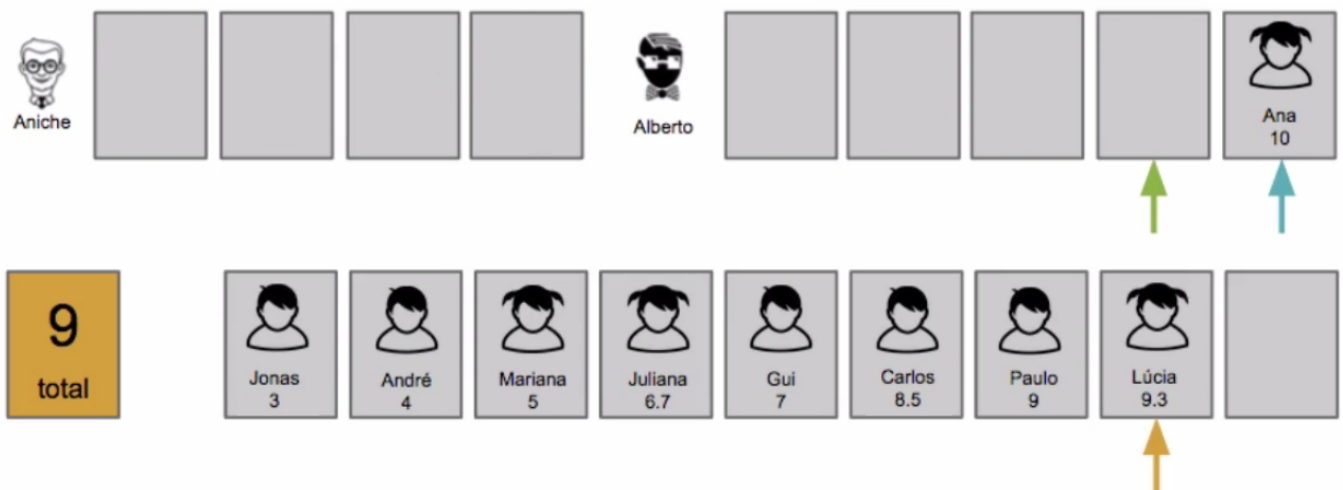


Seguimos para os próximos alunos. Quem se saiu melhor na prova, o Carlos ou a Lúcia? A Lúcia. Logo, é o Carlos que irá descer para o *array*.

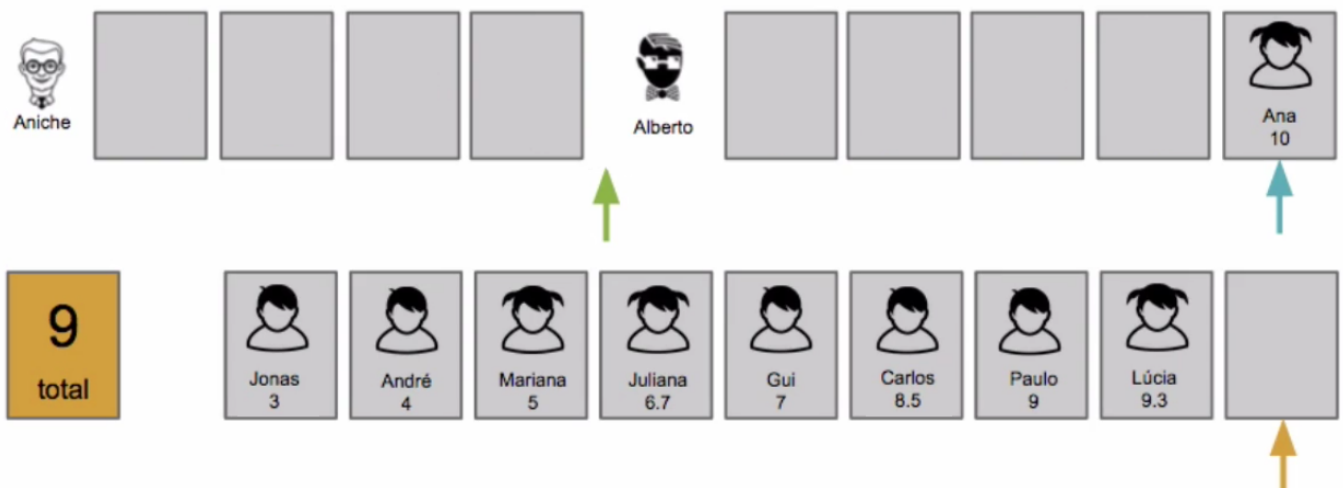


Comparando o Paulo e Lúcia, quem se saiu pior na prova? O Paulo. Vamos desce-lo para o *array*.

Então, um por um, fui comparando os alunos e o menor, foi sendo colocado no *array*. Com os dois alunos restantes, não faz mais sentido procurar o menor. Agora que já descemos todos os alunos de um dos grupos, vamos levar as duas restantes para o *array*. Colocaremos a Lúcia...



E depois, a Ana.





No fim, quando terminamos as comparações, sobraram duas alunas do grupo do Alberto. Pegamos os dois elementos e descemos para a lista única.

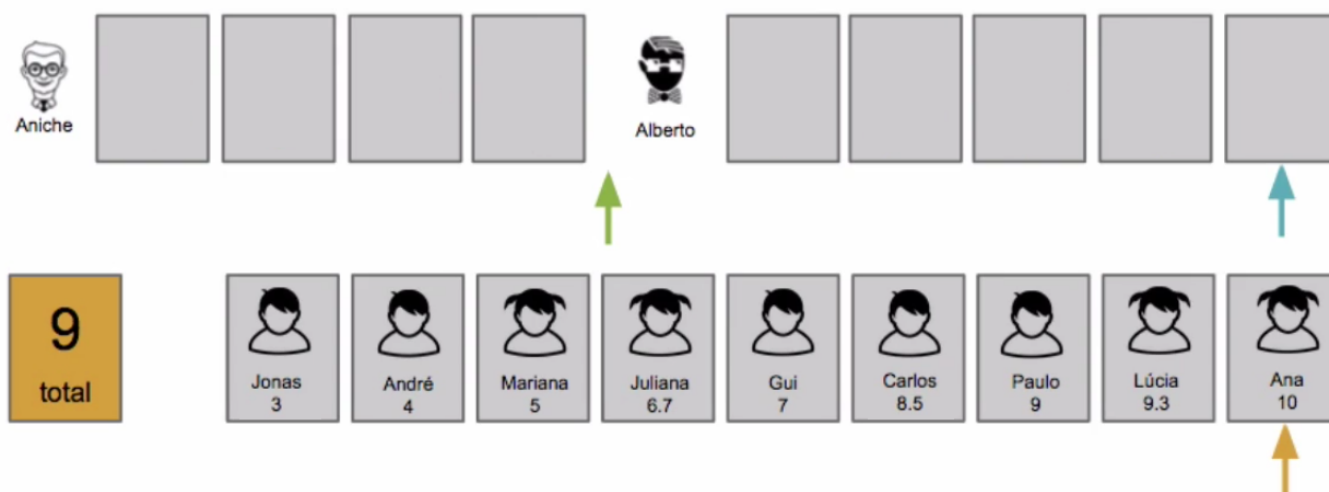


O que fizemos? Comparamos o elemento de um grupo com o de outro e identificamos qual era o menor. O maior permanecia e o menor descia para o novo *array*. Seguimos comparando as notas dos alunos e os menores eram colocados na lista abaixo. No fim, temos um *array* quase completo, porém sobraram algumas alunas no grupo do Alberto. Nós simplesmente descemos as duas e o *array* ficou ordenado.

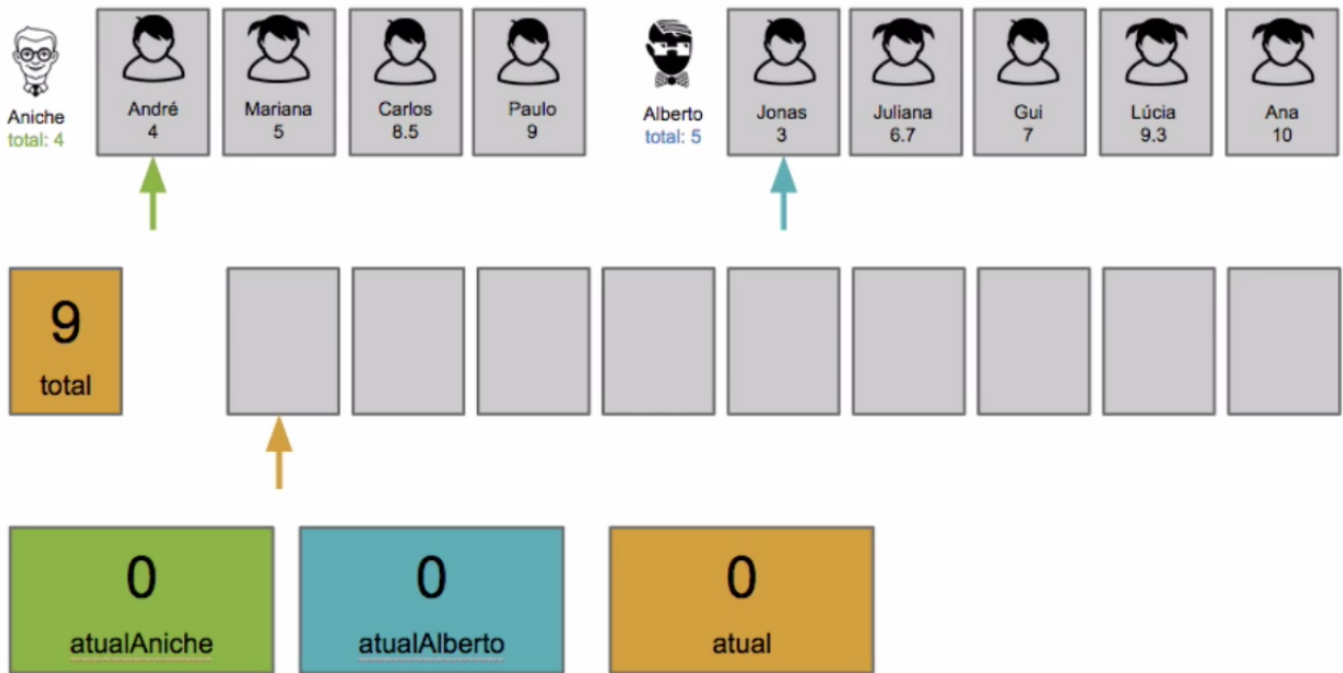
Com isto, na próxima vez que formos jogar baralho e todas as cartas terminarem embaralhadas, já sabemos como organizar: dividimos o monte em partes e distribuímos para os participantes ordenarem com seus próprios algoritmos. No fim, analisamos os grupinhos de cartas e comparamos os itens: "A menor carta é minha, coloca no monte. Agora é a sua, coloca no monte". As menores cartas serão agrupadas em um novo monte, até que não seja mais possível compará-las e as cartas estão ordenadas.

## Simulando com as variáveis

Então já sabemos intercalar dois *arrays* ordenados. Agora, quais variáveis eu preciso para ver isto acontecer?

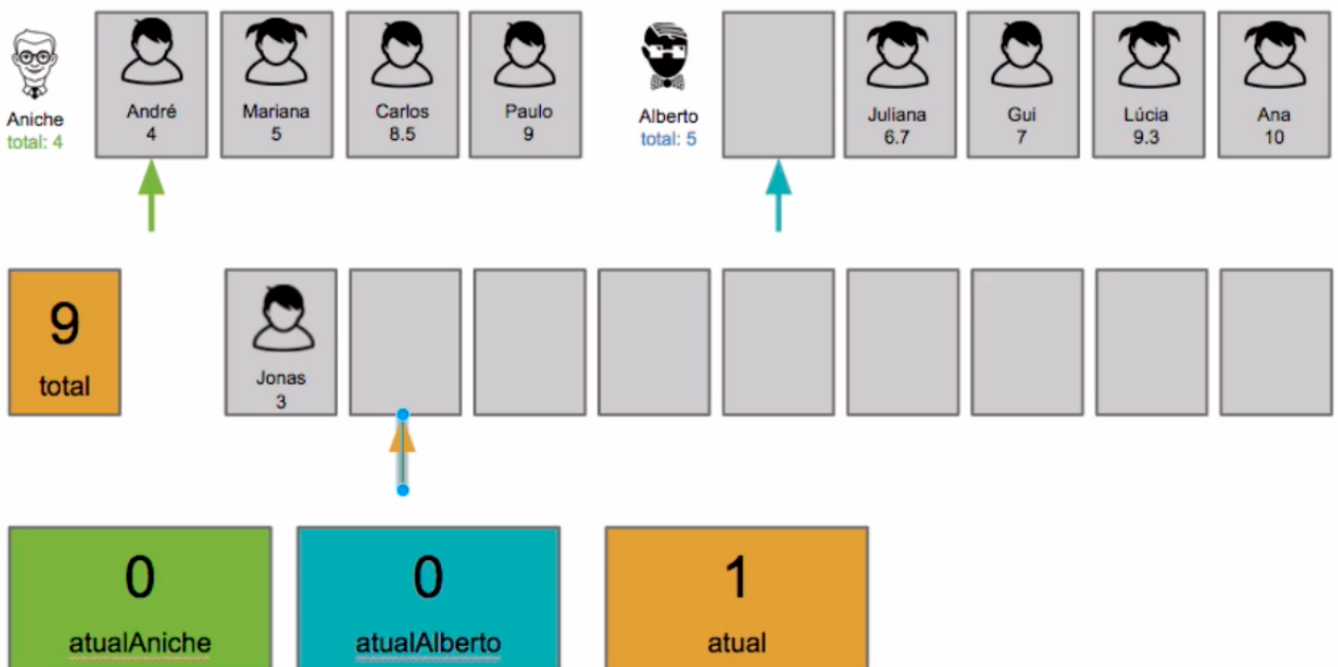


Nós utilizamos três setinhas, três variáveis, que nós brincamos e mudamos os valores. Além da variável referente ao total de elementos, que precisávamos saber desde o princípio.



Com o total, nós criamos o tamanho do nosso *array*, precisamos descobrir qual era o aluno **atual** do Aniche e qual era o do Alberto. Também precisávamos saber o número total de elementos em cada um dos grupos e assim vamos brincando com o atual das três listas, incluindo a geral. Como farei isto? Vamos simular de novo o algoritmo que usamos no exemplo, mas agora usaremos as variáveis.

Quando comparamos as notas dos alunos André e Jonas, qual dos dois teve o pior resultado? O Jonas. Então, iremos movê-lo para a posição atual do *array* geral. Em seguida, o que devemos fazer? Vamos somar 1 com a posição atual.



Faremos o mesmo com a atual do Alberto, porque queremos analisar o próximo elemento.