

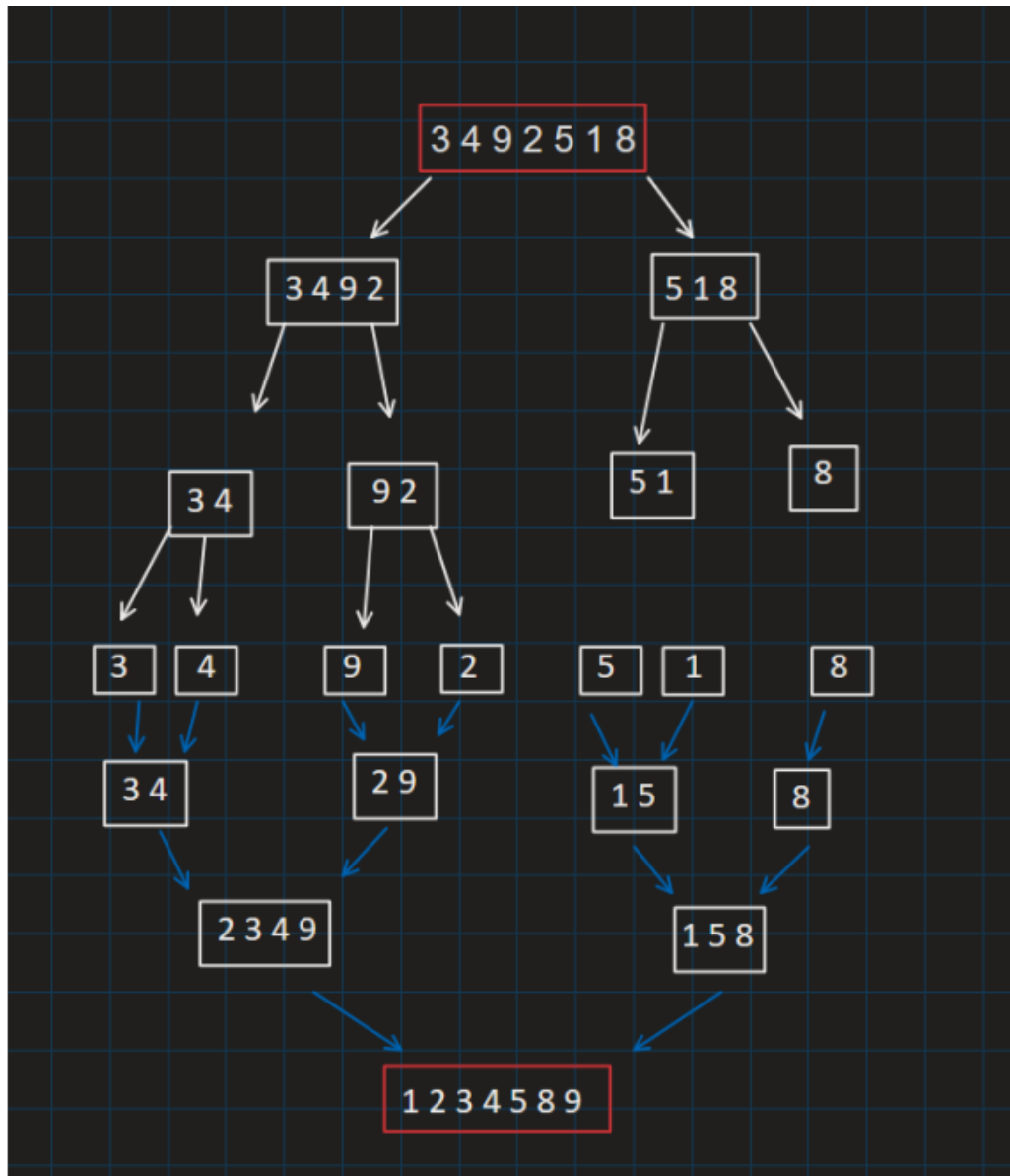
Atividade Métodos Sofisticados de Ordenação

1. Mostre um exemplo de entrada que demonstra que o ShellSort não é estável.

R: 1, 2, 4, 8, 16, ... É instável pois não compara elementos em posições pares com elementos em posições ímpares até a última iteração.

2. Dada a sequência de números: 3 4 9 2 5 1 8. Ordene em ordem crescente utilizando o algoritmo MergeSort, apresentando a sequência dos números a cada passo (teste de mesa).

R:

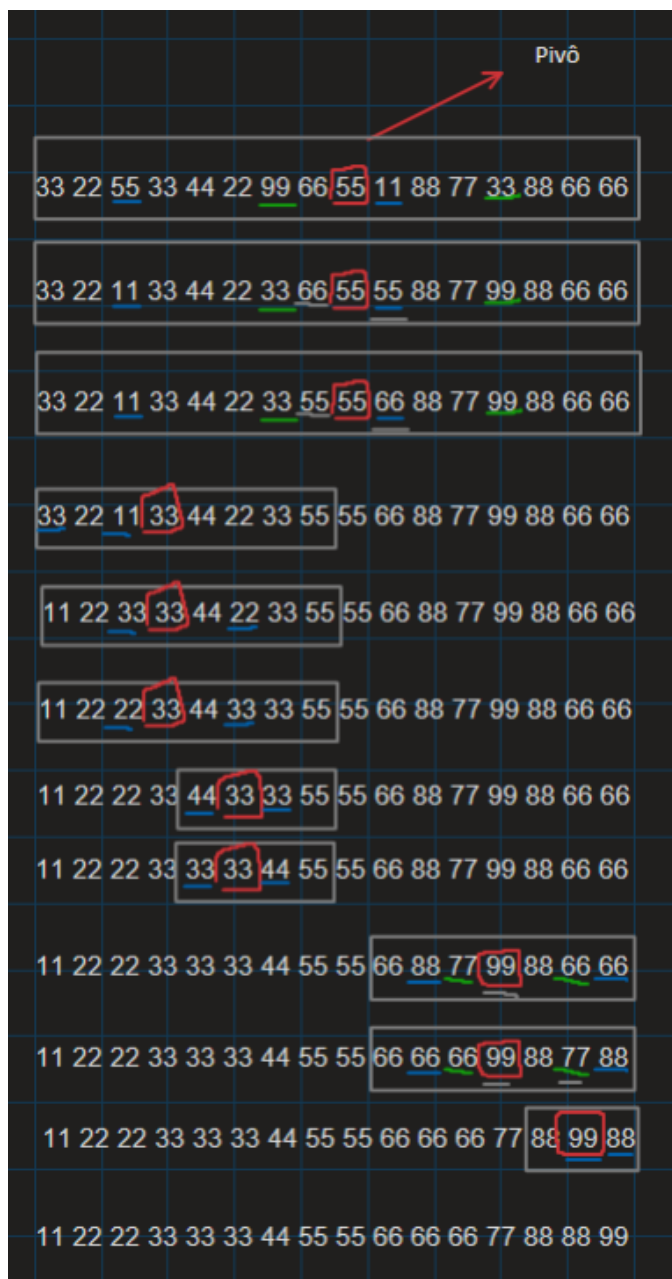


3. Mostre o resultado da operação $\text{particao}(v, 0, 15)$ do QuickSort para o vetor a seguir.

Utilize o elemento $v[\text{meio}]$ como sendo o pivô em cada chamada da função.

33 22 55 33 44 22 99 66 55 11 88 77 33 88 66 66

R: Na maioria dos casos do QuickSort, o número usado como pivô é o do meio, nesse caso é o 55, e a partir dele, o algoritmo analisa se os números à esquerda são menores que o da direita, se sim, ele continua do mesmo jeito, se não eles trocam de posição, quando essa etapa for concluída e os números à esquerda do pivô forem todos menores que o da direita, um novo pivô a direita do antigo é escolhido, com a finalidade de organizar os menores entre si, e assim sucessivamente, quando os números da esquerda estão organizados, ele faz o mesmo processo para os números da direita, até que todos fiquem ordenados propriamente.



4. Escreva uma função que decida se um vetor v é ou não um heap.

5. Use o algoritmo HeapSort para ordenar o vetor 16 15 14 13 12 11 10 9 8 7 6 5 4.

Mostre o estado do vetor no início de cada iteração.