1) Escolhido: Quicksort

4) tempos médios de execução Quicksort:

50mil números: 6ms

1000mil números: 12ms

5) tempos médios de execução Mergesort:

50mil números: 7ms

1000mil números: 17ms

6) int vetor[10] = {9,8,7,6,5,4,3,2,1,0}

Primeiro o algoritmo seta o pivô, no meio do vetor, neste caso o valor “5”. Começa verificando da esquerda para a direita e procura o primeiro valor maior que o pivô (9) e depois da direita para a esquerda, procurando o primeiro valor menor que o pivô (0). Após encontrados, caso o valor maior esteja posicionado antes do menor, troca eles de lugar, deixando o vetor = {0,8,7,6,5,4,3,2,1,9}. Executa novamente procurando o próximo maior e menor, até que o index dos dois se encontre.

Depois da primeira ordenação, divide o vetor em dois, um antes e um depois do pivô, e repete o processo para eles, dividindo-os e executando novamente até estarem na ordem correta.