**Aula 06 - Comparação em desempenho dos métodos de ordenação Bubble Sort, Insertion Sort, Selection Sort, Merge Sort e Quick Sort**

Após nossos estudos sobre os cinco métodos de ordenações ficam evidente que o método Quick Sort e o Merge Sort são os melhores, conforme é ilustrado nas figuras (1, 2, 3, 4 e 5) abaixo conseguimos identificar que sua quantidade de comparações e quantidade de trocas diminui significativamente.

****

Figura 1 - Gráfico de desempenho do método Bubble Sort

****

Figura 2 - Gráfico de desempenho do método Insertion Sort

  

Figura 3 - Gráfico de desempenho do método Selection Sort

****

Figura 4 - Gráfico de desempenho do método Merge Sort

****

Figura 5 - Gráfico de desempenho do método Quick Sort

**Atividade:**

Baseado na função abaixo que gera números ordenados, invertidos e aleatórios, construa um programa para simular os resultados das cinco figuras acima, deve ser simulando em três vetores de dez mil dados do tipo inteiro:

void geraNumero(int[] vet, int op) {

int i, j;

switch (op) {

case 1: // Ordenados

for (i = 0; i < vet.length; i++) {

vet[i] = i + 1;

}

break;

case 2: //Invertidos

for (i = 0; i < vet.length; i++) {

vet[i] = vet.length-i;

}

break;

case 3: // Aleatórios

for (i = 0; i < vet.length; i++) {

vet[i] = (int) (rand() % 1000);

}

break;

}

}

R.: