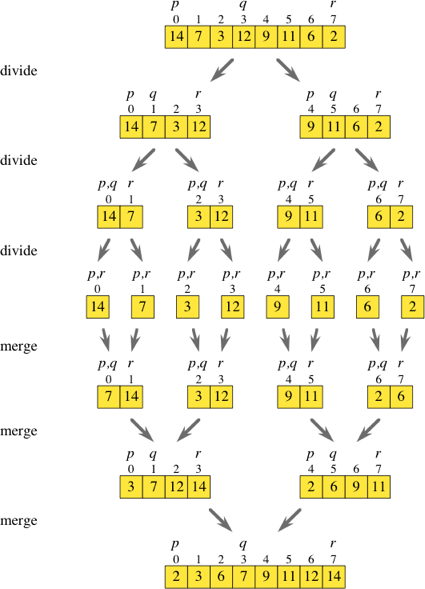
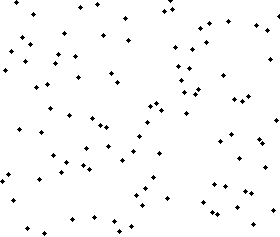
MergeSort

Criado por John von Neumann em 1945, este algoritmo utiliza a técnica de dividir para conquistar, diferentemente do seu primo, o algoritmo Bubble Sort. Possui complexidade de tempo Θ(n log 2n), no pior caso e no melhor caso, Θ(n log n).

O algoritmo Merge Sort é de fácil implementação. É conceitualmente mais fácil do que a ordenação quicksort e a ordenação Shell mas, possui uma desvantagem. Requer um vetor adicional na memória, igual em tamanho àquele sendo ordenado. Se seu vetor original mal couber na memória, a ordenação não funcionará. Porém, se você possuir bastante espaço, isso não será um problema.

Sua estratégia consiste em criar uma sequência ordenada a partir de outras duas já ordenadas. Para tal, divide-se a sequência original em pares de dados, e ordena-se. Depois, agrupa-se em sequências de quatro elementos, e assim por diante até a sequência original estar separada em apenas duas partes. Este processo se repete até o array estar totalmente unido e ordenado.

Sua ideia básica consiste em **Dividir** (o problema em vários subproblemas e resolver esses subproblemas através da recursividade) e **Conquistar** (após todos os subproblemas terem sido resolvidos ocorre a conquista que é a união das resoluções dos subproblemas). Como o algoritmo Merge Sort usa a recursividade, há um alto consumo de memória e tempo de execução, tornando esta técnica não muito eficiente em alguns problemas.



**Observações**

É possível implementar o merge sort utilizando somente um vetor auxiliar ao longo de toda a execução, tornando assim a complexidade de espaço adicional igual

É um algoritmo estável na maioria das implementações, em que elas podem ser iterativas ou recursivas.

É possível também implementar o algoritmo com espaço adicional

**Desvantagens**

Utiliza funções recursivas;

Gasto extra de memória. O algoritmo cria uma cópia do vetor para cada nível da chamada recursiva, totalizando um uso adicional de memória igual a