MAC121 - Algoritmos e Estruturas de Dados I

Universidade de São Paulo

Segundo Semestre de 2020

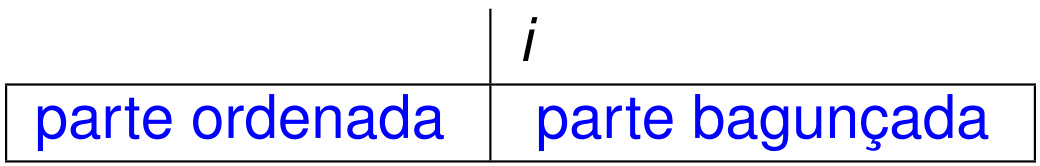
Ordenação por inserção

Mergesort

Ordenação por inserção

Ideia: Em cada iteração, o vetor está dividido em duas partes: uma primeira parte ordenada e a segunda, bagunçada. O primeiro elemento da parte bagunçada é colocado na posição correta na parte ordenada.

Invariante: No final da iteração i, o pedaço inicial do vetor, com i elementos, está ordenado.



Para colocar o primeiro elemento x da parte bagunçada no lugar, vamos movendo para frente todos os elementos da parte ordenada que são maiores que x, abrindo, assim, espaço para o novo elemento.

Ordenação por inserção

Qual o consumo de tempo? Quantas comparações e quantas movimentações?

E, se fizermos busca binária para decidir o lugar para colocar o elemento na parte ordenada? Este algoritmo é chamado de inserção binária.

O número de comparações é limitado por

$$\sum_{i=2}^{n-1} \log_2 i$$

Animação do algoritmo

Mergesort

O Mergesort é o primeiro algoritmo com consumo de tempo $O(n \log n)$. Foi inventado em 1945, pelo matemático John von Neumann.

É baseado na ideia de Divisão e Conquista.

Separe o vetor na metade e ordene-as separadamente. Intercale, então, os vetores ordenados, obtendo o vetor inteiro ordenado.

44	75	12	27	95	−15	3	73	19	8	12	22

−15	12	27	44	75	95

3	8	12	19	22	73

−15	3	8	12	12	19	22	27	44	73	75	95

Intercalando vetores ordenados

Problema: Considere um vetor v e três índices p, q e r do vetor, tais que, o intervalo $[p, q-1]$ e $[q, r-1]$ estão ordenados. Obtenha o vetor ordenado no intervalo $[p, r-1]$.

```
vamos usar um vet auxiliar para guardar o resultado

enquanto houver itens nas duas partes do vetor
  compara os menores elementos das duas partes
  que ainda não foram incluídos, e inclui no vetor
  auxiliar o menor

verifica se há elementos ainda a incluir
copia do vetor auxiliar de volta para o original
```

Mergesort

Implementando o Mergesort. Qual o consumo de tempo?

Se $T(n)$ é o tempo que o mergesort leva para ordenar um vetor com n elementos, temos:

$$\begin{aligned} T(n) &= T(\tfrac{n}{2}) + T(\tfrac{n}{2}) + n \\ &= 2T(\tfrac{n}{2}) + n \\ &= 2(2T(\tfrac{n}{4}) + \tfrac{n}{2}) + n \\ &= 4T(\tfrac{n}{4}) + 2n \\ &= 4(2T(\tfrac{n}{8}) + \tfrac{n}{4}) + 2n \\ &= 8T(\tfrac{n}{8}) + 3n \\ &\vdots \\ &= 2^k T(\tfrac{n}{2^k}) + kn. \end{aligned}$$

Tomando $k = \log n$, obtemos que $T(n) = O(n \log n)$.

Animação do algoritmo

Para saber mais

- Material sobre ordenação por inserção (P. Feofiloff)
- Material sobre Mergesort (P. Feofiloff)
- Livro texto, capítulos 8 e 9