

Aulas de Estruturas de Dados e algoritmos 2

cotrim149

August 19, 2014

1 Busca Sequencial

1. Complexidade média(Tempo de demora para resposta):

$$\frac{n}{2} \quad (1)$$

2. $O(n)$

3. Métodos para otimização

- Sentinela: Consiste em adicionar um elemento de valor x no final do vetor.

4. Alternativa: Lista encadeada

5. Aumento de eficiência

- Método mover para frente: Sempre que uma pesquisa obter êxito, o registro recuperado é colocado no início da lista. **Desvantagem:** Qualquer informação fica privilegiada
- Método da transposição: Um registro recuperado com sucesso é trocado imediatamente com o elemento anterior (swap é $O(1)$, não importando a quantidade de elementos). **Desvantagem:** Cancelamento da otimização, (swap alternados entre mesmos elementos)

6. Tabela Ordenada

- Complexidade: $O(n/2)$. **Pior caso:** Complexidade: $O(n)$
- Dificuldade: Manter tabela ordenada e a ordenação em si

7. Tabela indexada

- Utilização de tabela auxiliar como tabela de índices
- Cada elemento na tabela de índices contém uma chave (kindex) e um indicador do registro no arquivo que corresponde a kindex

8. Vantagens e desvantagens na busca sequencial

- Vantagens: Os itens poderão ser examinados sem serem acessados, o tempo de busca diminui consideravelmente
- Desvantagens: Tabela tenha que estar ordenada, demanda mais espaço.

9. Remoção

- Remova-se o elemento e rearranja-se a tabela
- Indicar que o local está vazio, e futuramente é inserido outro elemento no índice

10. Inserção

- Se houver espaço vago, rearranjam-se os elementos localmente, caso não haja espaço, toda a tabela deve ser rearranjada

2 Busca Binária

- $O(\log n)$; Cada comparação reduz o número de possíveis candidatos por um fator de 2.
- Pode ser usada como organização de tabela sequencial indexada
- Desvantagem: Precisa de índices, não funciona em uma lista encadeada ou duplamente encadeada

3 Busca por interpolação

- As chaves precisam estar uniformemente distribuídas

$$meio = inf + (sup - inf) * \frac{(x - A[inf])}{(A[sup] - A[inf])} \quad (2)$$

- $O(\log(\log(n)))$ se as chaves estiverem uniformemente distribuída
- Se chaves não estiverem uniformemente distribuídas, a busca por interpolação pode ser tão ruim quanto uma busca sequencial
- Desvantagem: Em situações práticas, as chaves tendem a se aglomerar em torno de determinados valores e não são uniformemente distribuídas

4 Busca em árvore

1. pré-ordem (sempre a esquerda): [8,3,1,6,4,7,10,14,13]
2. in-ordem (sempre em baixo): [1,3,4,6,7,8,10,13,14]
3. pós-ordem (sempre a direita) : [1,4,7,6,3,13,14,10,8]

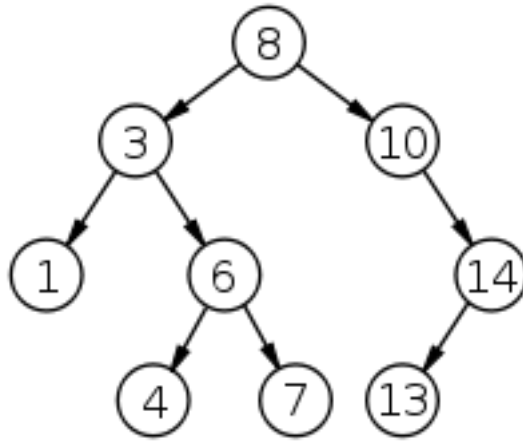


Figure 1: Imagine uma linha passando sempre a esquerda de cada valor começando pela raiz

5 Selection Sort

- Chamado de algoritmo natural
- Se baseia em passar o menor valor do vetor para a primeira posição
- complexidade média = $(n-1) * (n/2)$
- $O(n^2)$

6 Insertion Sort

- Chamado de algoritmo natural
- Não existe swap para fora do vetor, sempre acrescenta o "procurado" entre os valores
-

$$complexidadeMedia = (n) * \left(\frac{n}{4}\right) \quad (3)$$

$$Complexidade = O(n^2) \quad (4)$$

7 Bubble sort

$$Complexidade = O(n^2) \quad (5)$$

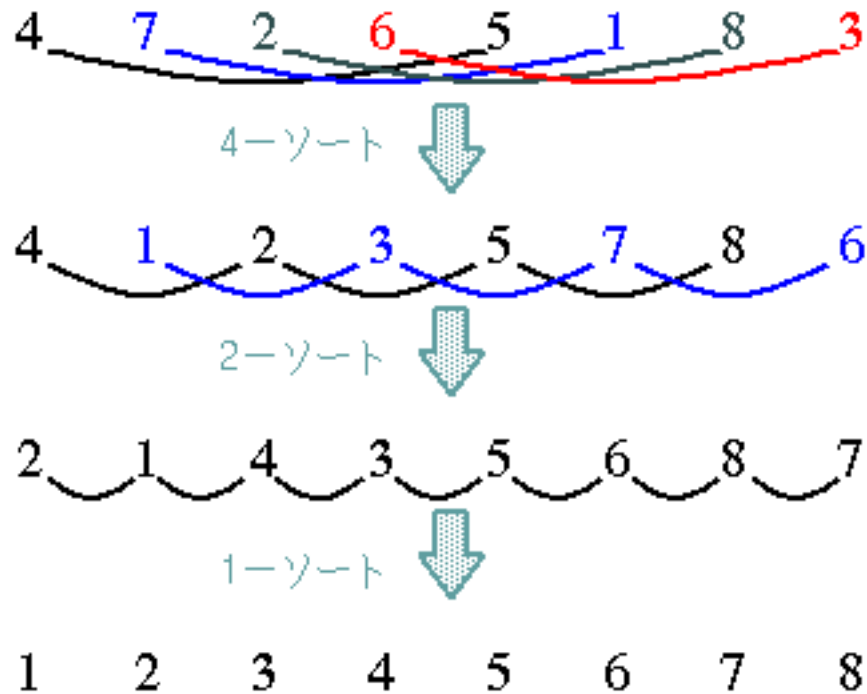


Figure 2: **Shell Sort** Linha 1: Gap de $n=4$ com 4 subvetores; Linha 2: Gap de $n=2$ com 2 subvetores; Linha 3: Gap de $n=1$ com 1 subvetor, Linha 4: Ordenação completa

8 Shell sort

- O mais eficiente algoritmo de ordenação dentre os de complexidade quadrática
- "Gaps" de $n/2$, sempre fazer o arredondamento para baixo.
- Vizinho é igual a um "gap" de distância
- Refinamento do insertion Sort
- subvetores dentro do vetor(ideal!), subvetor com tamanho $n/\text{"gap"}$

$$\text{Complexidade} = O(n^2) \quad (6)$$

9 Bucket Sort

- "Dividir e conquistar"

- Divisão em "baldes" por faixas de valores,
- pode ser usado qualquer algoritmo dentro de cada balde

10 Quick Sort

- "Dividir e conquistar"
- Escolhe-se um valor pivô e move-se todos os valores menores para a esquerda e os maiores para a direita
- Ordena-se recursivamente os valores menores e maiores
- Algoritmo instável: Não garante que elementos iguais não invertam, não se preocupa com a ordem dos elementos
- $O(\log(n))$
- Desvantagem: A escolha de um mau pivô seguidamente podem tornar o algoritmo $O(n^2)$
- Encontrar o mediano é $O(n)$, o que resulta $O(n \log(n))$