

ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES II

Algoritmos de busca

Busca

Nas últimas aulas vimos vários algoritmos de ordenação.

Nesta aula, veremos uma das aplicações de ordenação, que é na busca.

Ordenar um conjunto antes de realizar a busca permite melhorar o desempenho do algoritmo.

Busca Linear

Em Python, a busca por um elemento em uma lista pode ser feita de duas formas:

- Com o in:

```
>>> l = [7, 6, 3, 4]
```

```
>>> 3 in l
```

```
True
```

```
>>> 8 in l
```

```
False
```

- Com o index():

```
>>> l = [7, 6, 3, 4]
```

```
>>> l.index(3)
```

```
2
```

```
>>> l.index(5)
```

```
→ Erro
```

Busca Linear

Importante notar que se a lista não estiver ordenada, a única maneira de implementar o operador `in` e o método `index()` é por meio de uma "varredura" item a item na lista até encontrá-lo.

No pior caso, o algoritmo percorre toda a lista, ou seja, seu tempo de execução é proporcional ao tamanho da lista.

Busca Binária

Se os elementos na lista são comparáveis é possível ordená-los inicialmente, usando algum método de ordenação (e.g. método `sort()`):

```
>>> l.sort()
```

```
>>> l
```

```
[3, 4, 6, 7]
```

Com isso, podemos aplicar algoritmos de busca mais eficientes, como a busca binária.

Busca Binária

Ao invés de comparar inicialmente o elemento desejado com o primeiro elemento (índice 0), suponha que o comparamos com o elemento armazenado num índice i qualquer.

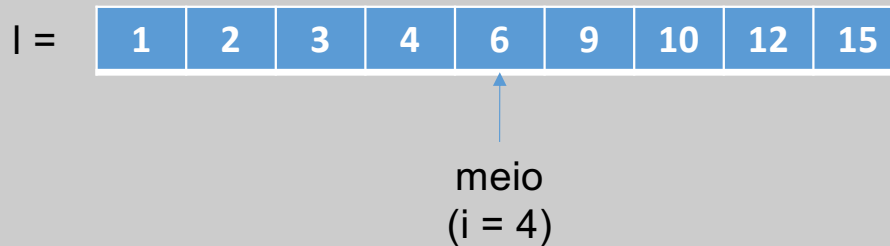
Teremos 3 possibilidades:

- $l[i] == \text{elemento desejado} \rightarrow$ finaliza a busca
- $l[i] > \text{elemento desejado} \rightarrow$ elemento pode estar à esquerda
- $l[i] < \text{elemento desejado} \rightarrow$ elemento pode estar à direita

Busca Binária

Exemplo: selecionar o meio da lista

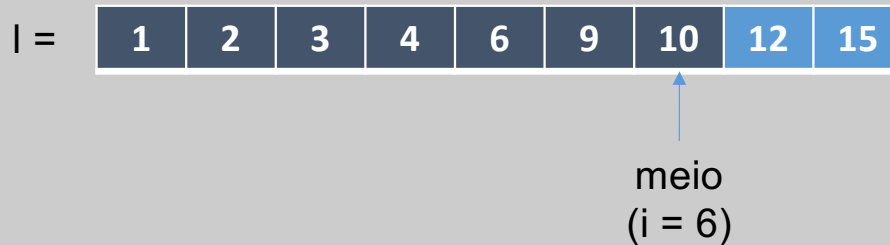
Elemento procurado: 12



Busca Binária

Exemplo: selecionar o meio da lista

Elemento procurado: 12



Busca Binária

Exemplo: selecionar o meio da lista

Elemento procurado: 12



Busca Binária

Exemplo: selecionar o meio da lista

Elemento procurado: 12

| =

1	2	3	4	6	9	10	12	15
---	---	---	---	---	---	----	----	----

Número de comparações:

- Busca linear: 8
- Busca binária: 3

Busca Binária

Implementação em Python de uma função recursiva que realiza a busca binária em uma lista.

ALGORITMOS E PROGRAMAÇÃO DE COMPUTADORES II

Algoritmos de busca