

* Depth bound DFS

```
def DFSearch(G, s, goal, max-depth)
```

```
    F = {<s>}
```

```
    hit-depth = False
```

```
    while F:
```

```
        <s1...uk> = F.pop(-1)
```

```
        if goal(uk):
```

```
            return {<s1...uk>, hit-depth}
```

```
        for <uk, u> ∈ A:
```

```
            if len(<s1...uk, u>) < max-depth:
```

```
                F.append(<s1...uk, u>)
```

```
                hit-depth = True
```

```
    return {<s>, hit-depth}
```

* Iterative Deepening

```
def IDSearch(G, s, goal, max-depth)
```

```
    depth-bound = 0
```

```
    hit-bound = True
```

```
    path = []
```

```
    while hit-bound:
```

```
        {path, hit-bound} = DFSearch(G, s, goal, depth-bound)
```

```
        depth-bound += 1
```

```
    return path
```

* Busca Informada (heurística)

- "Informação": Qual ^{nó} ^{estado} é o mais promissor?

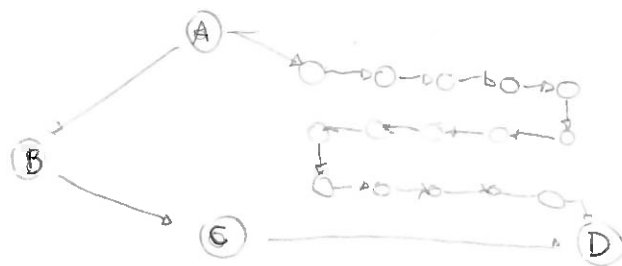
* Função heurística

$$h(n) : N \rightarrow \mathbb{R}_0^+$$

↳ Estimativa do menor custo possível entre n e algum objetivo.

$h(n)$ é uma heurística admissível se $h(n)$ é sempre menor ou igual ao custo do menor caminho entre n e algum objetivo

* Guloso



* A*

* Depth-first branch and bound