

16. 高级搜索 3：启发式搜索 Heuristic Search

- Heuristic: 智能地 或 根据某一项条件不断地优化搜索方向, 即, 一边搜索一边有所谓的思考在里面
- 本质: 通过优先级不断地去找要找的点, 先用优先级高的拿出来搜索即可

基于 BFS 代码

```
def BFS(graph, start, end):  
  
    queue = []  
    queue.append([start])  
    visited.add(start)  
  
    while queue:  
        node = queue.pop() # can we add more intelligence here?  
        visited.add(node)  
  
        process(node)  
        nodes = generate_related_nodes(node)  
        queue.push(nodes)
```

A * Search 代码模版

| 将 queue 改换成 priority_queue

```
from typing import Collection  
  
def AStarSearch(graph, start, end):  
  
    pq = collections.priority_queue() # 优先级 -> 估价函数  
    pq.append([start])  
    visited.append(start)
```

```
while pq:
    node = pd.pop() # can we add more intelligence here?
    visited.add(node)

    process(node)
    nodes = generate_related_nodes(node)
    unvisited = [node for node in nodes if node not in visited]
    pq.push(unvisited)
```

估价函数

启发式函数： $h(n)$ 它用来评价哪些节点最有希望的是一个我们要找的节点， $h(n)$ 会返回一个非负实数，也可以认为是从节点 n 的目标节点路径的估计成本。

启发式函数是一种告知搜索方向的方法，它提供了一种明智的方法来猜测哪个邻居节点会导向一个目标。

#Algorithm/Part II : Theory/Algorithm#