

INFORMED SEARCH STRATEGIES

Best-First Search

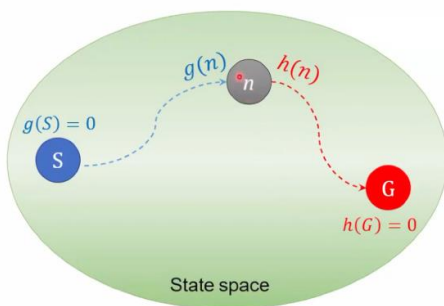
Có thể là TREE-SEARCH hoặc GRAPH-SEARCH

Tiêu chí chọn node dựa vào hàm đánh giá $f(n) = G$ (Nhỏ nhất thì lấy ra)

Heuristic function $f(n)$: Ước lượng chi phí ngắn nhất tại node n đến đích.

Ràng buộc để định nghĩa một hàm Heuristic hợp lệ:

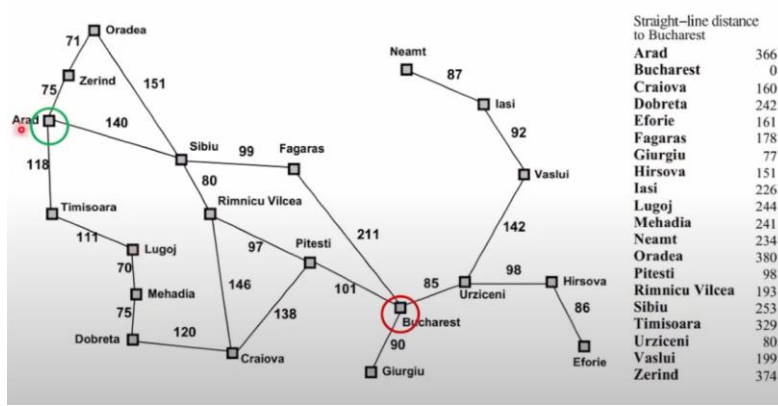
- + Không âm (Không quan tâm đường đi âm), tính đặc trưng cho mỗi bài toán.
- + n là goal, $h(n) = 0$



Greedy Best-First Search

Mở những node nào tỏ ra là gần GOAL nhất. Sử dụng $f(n) = h(n)$

Ví dụ minh họa 1: Bài các thành phố



(d) After expanding Fagaras

```

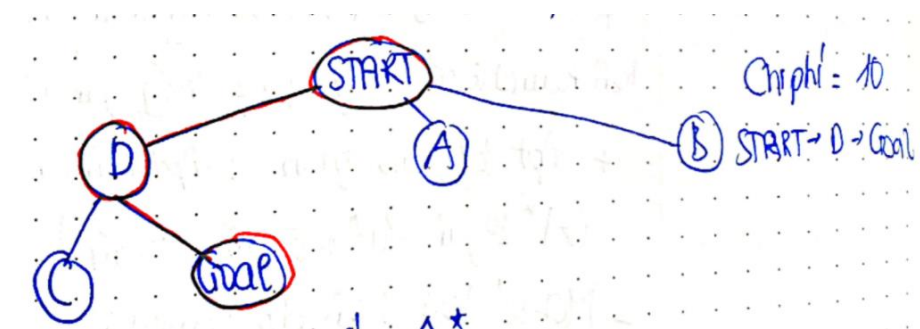
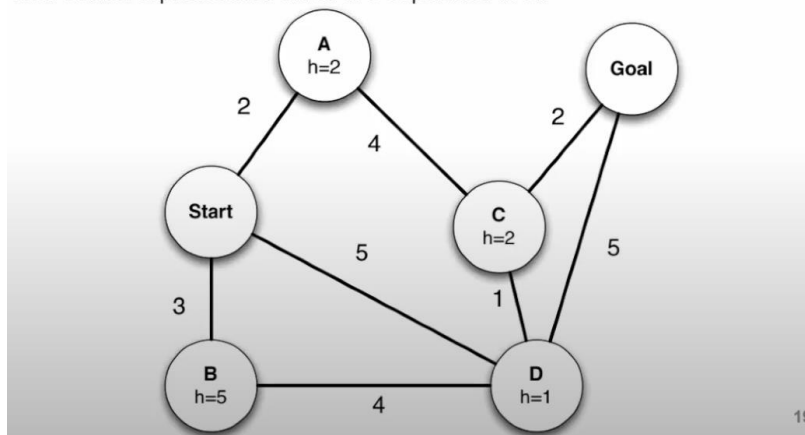
graph TD
    Arad --> Sibiu
    Arad --> Timisoara
    Arad --> Zerind
    Sibiu --> Arad
    Sibiu --> Fagaras
    Sibiu --> Oradea
    Sibiu --> Rimnicu_Vilcea[Rimnicu Vilcea]
    Fagaras --> Sibiu
    Fagaras --> Bucharest
  
```

Node	Cost
Arad	366
Bucharest	0
Craiova	160
Drobeta	242
Eforie	161
Fagaras	176
Giurgiu	77
Hirsova	151
Iasi	226
Lugoj	244
Mehadia	241
Neamt	234
Oradea	380
Pitesti	100
Rimnicu Vilcea	193
Sibiu	253
Timisoara	329
Urziceni	80
Vaslui	199
Zerind	374

Figure 17: A simplified road map of part of Romania. The map shows cities as nodes and roads as edges. The cost of each road is indicated by a number. The cost of the path from the root to the child is shown below the child node. The cost of the path from the root to the grandchild is shown below the grandchild node. The cost of the path from the root to the great-grandchild is shown below the great-grandchild node.

Ví dụ minh họa 2: Bài mở rộng theo bảng chữ cái

Work out the order in which states are expanded, as well as the path returned by graph search. Assume ties resolve in such a way that states with earlier alphabetical order are expanded first.



A* Search

Sử dụng Heuristic nhưng không phải nguồn thông tin duy nhất, tránh những đường đi có chi phí lớn -> Yên tâm sẽ nhận được đường đi ngắn nhất (cần kiểm tra tính hợp lệ $h(n)$)

$f(n) = g(n) + h(n)$ ($g(n)$: chi phí tới n , $h(n)$: chi phí ước lượng từ n đến G)

Một số điều kiện cho Heuristic:

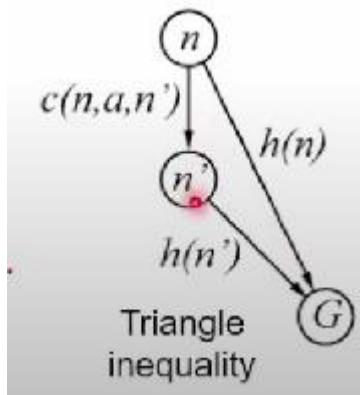
+ **Tính chấp nhận được:** Không bao giờ ước lượng quá chi phí đến đỉnh goal (nhỏ hơn chi phí bỏ ra thực sự (lời giải)) $h(n) \leq h^*(n)$. $h^*(n)$ không xác định giá trị cụ thể được, $h^*(n)$ được ước lượng bằng toán học.

Đối với giải thuật A* dùng TREE-SEARCH, $h(n)$ chấp nhận được thì đảm bảo tối ưu (Được phép mở lại đỉnh đã đi qua)

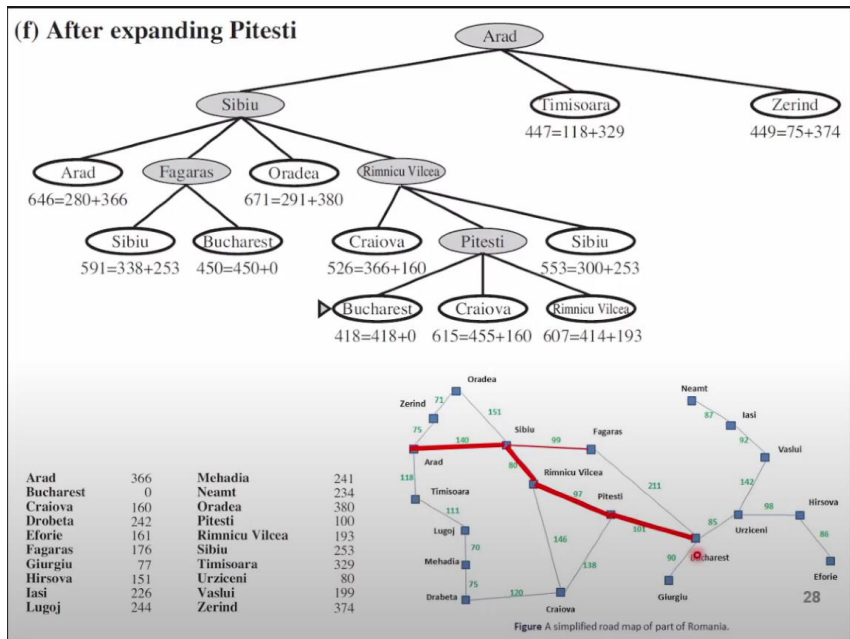
Đối với giải thuật A* dùng GRAPH-SEARCH, sử dụng tập đóng Close những đỉnh nào đã đi vào close không được phép đi ra nữa, $h(n)$ là 1 heuristic nhất quán.

Consistency heuristic (Heuristic nhất quán): Cũng chính là admissible heuristic.

Nếu tất cả các node n , mọi con n' của nó phát sinh bởi hành động A bất kì: $h(n) \leq c(n, a, n') + h(n')$

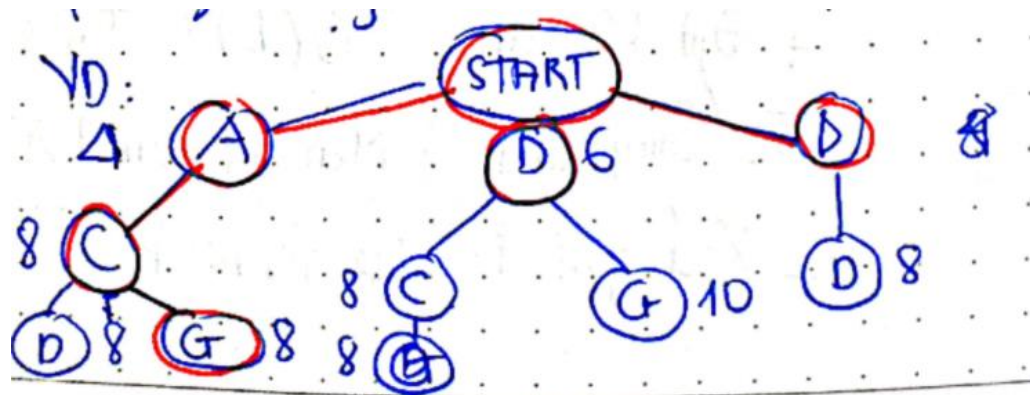


Ví dụ minh họa 1: Bài các thành phố



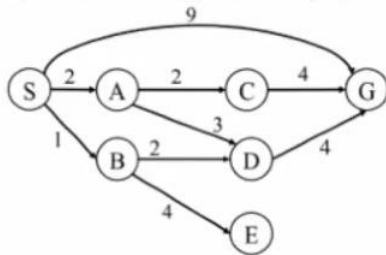
Thứ tự các đỉnh mở là: Arad(366) -> Sibiu(393) -> Rimnicu Vilcea(413) -> Fagaras(415) -> Pitesti(417) -> Bucharest(418)

Ví dụ minh họa 2: Bài mở rộng theo bảng chữ cái



Ví dụ minh họa 3:

Question 1 (1.0pt) Consider the following search problem, represented as a graph. The initial state is **S** and the only goal state is **G**. Assume ties resolve in such a way that states with earlier alphabetical order are expanded first. For each of the following search strategies, work out the list of expanded states as well as the path returned.



Heuristic						
S	A	B	C	D	E	G
6	0	6	4	1	10	0

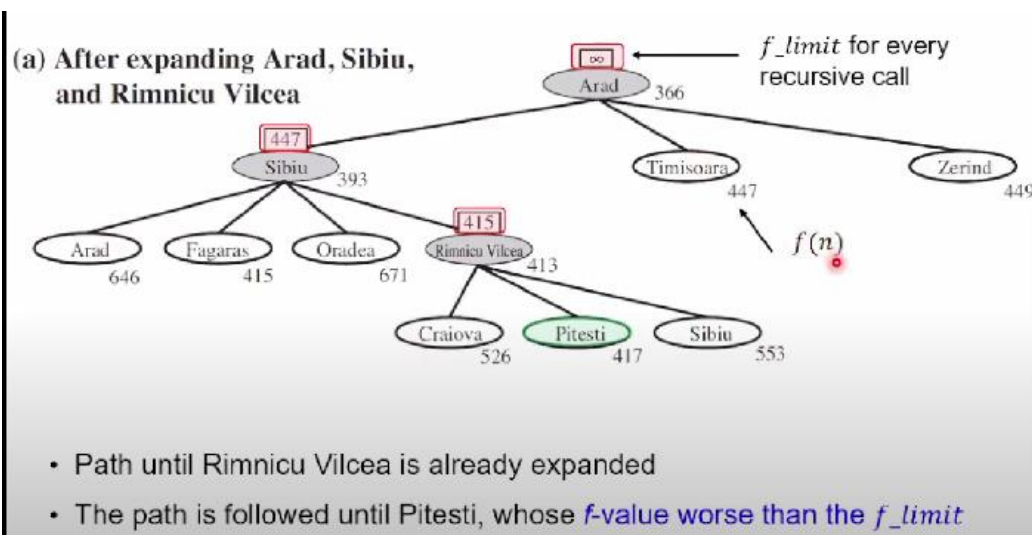
Algorithms	List of expanded states	Path returned
GBFS	S A G	S G
A*	S A D B C G S A D B D G	S A C G (graph search) S B D G (tree search)

Recursive Best-First Search (RBFS)

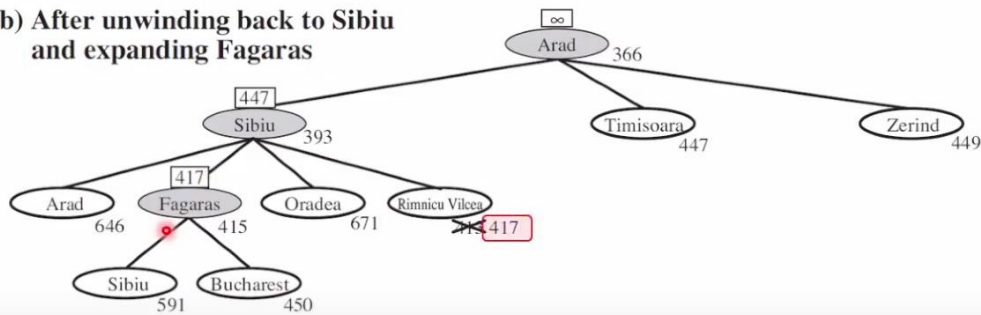
Lưu f -value của đường đi thay thế tốt nhất của bất kì nút tổ tiên nào của nút đang đứng. Quay lui nếu giá trị nút đang đứng $> f_limit$.

Trong quá trình quay lui, thay thế giá trị f_value của mỗi node bằng giá trị đường đi tốt nhất của các node con của nó.

Ví dụ minh họa: Bài các thành phố

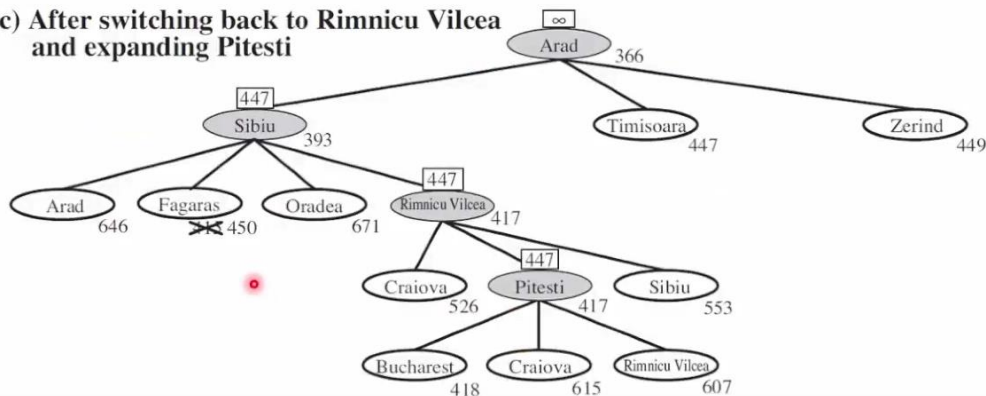


(b) After unwinding back to Sibiu and expanding Fagaras



- Unwind recursion and store best f -value for current best leaf Rimnicu Vilcea
 - $result, best.f \leftarrow RBFS(problem, best, \min(f_limit, alternative))$
- $best$ is now **Fagaras**. Call RBFS for new $best$
 - $best$ value is now 450

(c) After switching back to Rimnicu Vilcea and expanding Pitesti



- Unwind recursion and store best f -value for current best leaf of Fagaras
 - $result, best.f \leftarrow RBFS(problem, best, \min(f_limit, alternative))$
- $best$ is now **Rimnicu Viclea** (again). Call RBFS for new $best$
 - Subtree is again expanded
 - Best alternative subtree is now through Timisoara
- Solution is found since because $447 > 418$.