

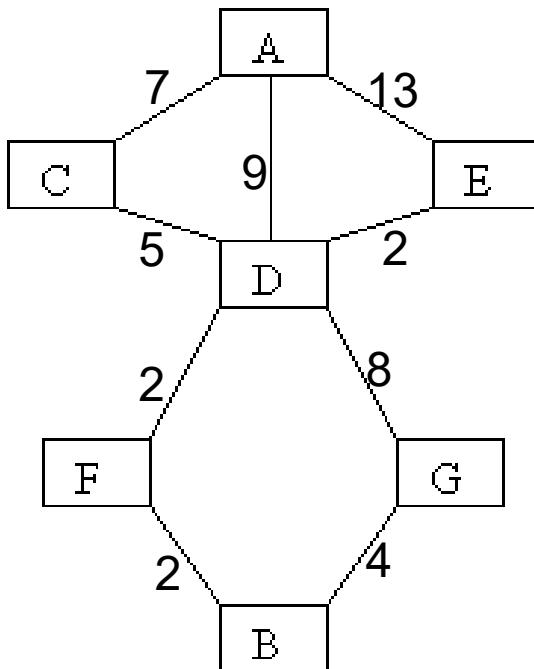
Tìm kiếm tối ưu

TS. Nguyễn Quốc Tuấn

Bộ môn Mạng và Các HTTT

Bài toán tìm đường đi ngắn nhất:

- Cho một không gian trạng thái
 - Giả sử chi phí để đưa u thành v là $k(u,v)$
 - Tìm đường đi từ trạng thái đầu tới trạng thái đích, sao cho tổng chi phí là nhỏ nhất

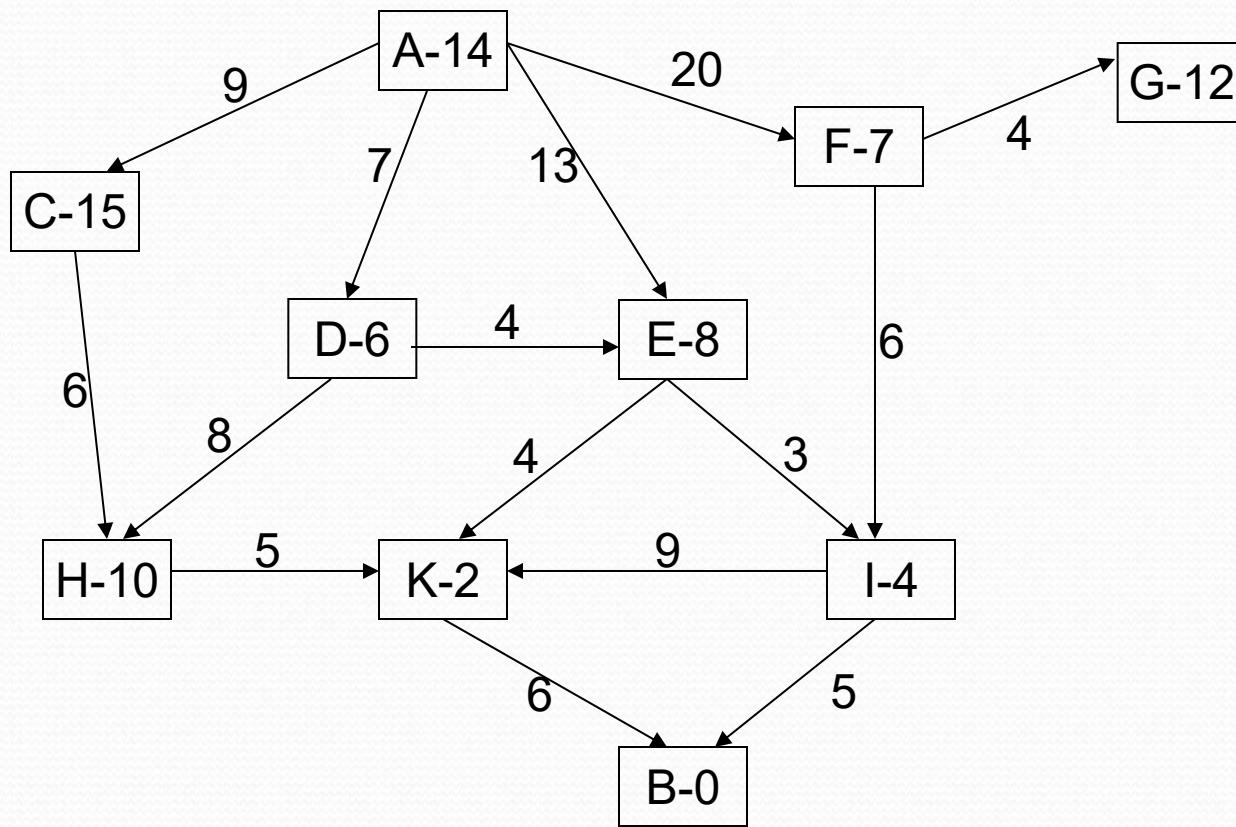


- Ví dụ: Bài toán tìm đường đi trên bản đồ giao thông. Ta cần tìm đường đi ngắn nhất từ A đến B
 - Tìm kiếm A*, và Tìm kiếm nhánh và cận
 - Tìm kiếm kinh nghiệm
 - Hàm đánh giá?

Hàm đánh giá trong tìm kiếm tối ưu

- Giả sử u là một trạng thái đạt tới (có đường đi từ trạng thái đầu u_0 tới u) ta xây dựng 2 hàm đánh giá:
 - $g(u)$ là đánh giá độ dài đường đi ngắn nhất từ u_0 đến u
 - $h(u)$ là đánh giá độ dài đường đi ngắn nhất từ u đến trạng thái đích
 - Có thể dùng leo đồi với hàm h -Không tìm được đường đi ngắn nhất
 - Có thể dùng tốt nhất đầu tiên với hàm h -Tìm được đường đi ngắn nhất, tuy nhiên có thể kém hiệu quả
- Giải pháp:
 - Sử dụng hàm đánh giá $f(u)=g(u)+h(u)$
 - $f(u)$ là đánh giá độ dài đường đi ngắn nhất qua u từ trạng thái đầu tới trạng thái kết thúc

- Cho đồ thị không gian trạng thái, TTĐ: A, TTKT: B



Thuật toán A*

Sử dụng kỹ thuật tìm kiếm tốt nhất đầu tiên với hàm đánh giá $f(u)$, tức là tìm kiếm theo bề rộng dưới sự hướng dẫn của hàm đánh giá.

Procedure A*

Begin

Khởi tạo danh sách L chỉ chứa trạng thái ban đầu;

2. Loop do

2.1. if L rỗng then { thông báo thất bại; stop}

2.2. Loại trạng thái u ở đầu danh sách L

2.3. if u là trạng thái kết thúc then {thông báo thành công; stop}

2.4. for mỗi trạng thái v kề u do

{ $g(v)=g(u)+k(u,v);$

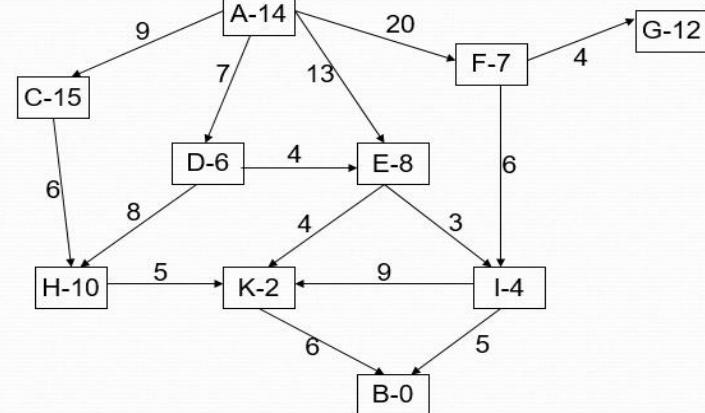
$f(v)=g(v)+h(v);$

xen v vào danh sách L sao cho L được sắp theo thứ tự tăng dần của f}

End;

Thuật toán A*

Mô tả quá trình tìm kiếm



TT	TTK	$k(u,v)$	$h(v)$	$g(v)$	$f(v)$	Danh sách L
A	C	9	15	9	24	D13,E21,C24,F27
	D	7	6	7	13	
	E	13	8	13	21	
	F	20	7	20	27	
D	H	8	10	15	25	E19,E21,C24,H25,F27
	E	4	8	11	19	
E	K	4	2	15	17	K17,I18,E21,C24,H25,F27
	I	3	4	14	18	
K	B	6	0	21	21	I18,B21,E21,C24,H25,F27
I	K	9	2	23	25	B19, B21,E21,C24,H25,K25,F27
	B	5	0	19	19	
B	TTKT/dừng, đường đi $B \leftarrow I \leftarrow E \leftarrow D \leftarrow A$, độ dài 19					

Thuật toán Tìm kiếm Nhánh và cận

Sử dụng kỹ thuật tìm kiếm leo đồi với hàm đánh giá $f(u)$, tìm kiếm theo chiều sâu dưới sự hướng dẫn của hàm đánh giá

Procedure TK-Nhanh&Can

Begin

Khởi tạo danh sách L chỉ chứa trạng thái ban đầu; gán giá trị đầu cho cost

2. Loop do

2.1. if L rỗng then {stop}

2.2. Loại trạng thái u ở đầu danh sách L

2.3. if u là trạng thái kết thúc then

if $g(u) \leq \text{cost}$ then { $\text{cost} = g(u)$; quay lại bước 2.1}

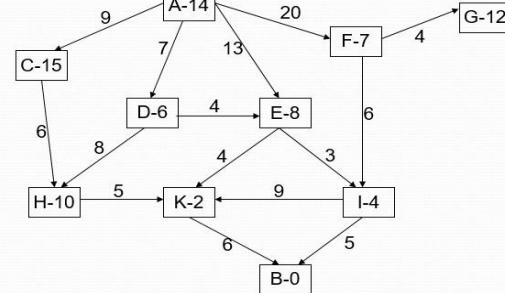
2.4. for mỗi trạng thái v kề u do

$\{ g(v) = g(u) + k(u,v); f(v) = g(v) + h(v);$

xen v vào danh sách L1 sao cho L1 được sắp theo thứ tự tăng dần của f}

2.5. Chuyển L1 vào đầu danh sách L

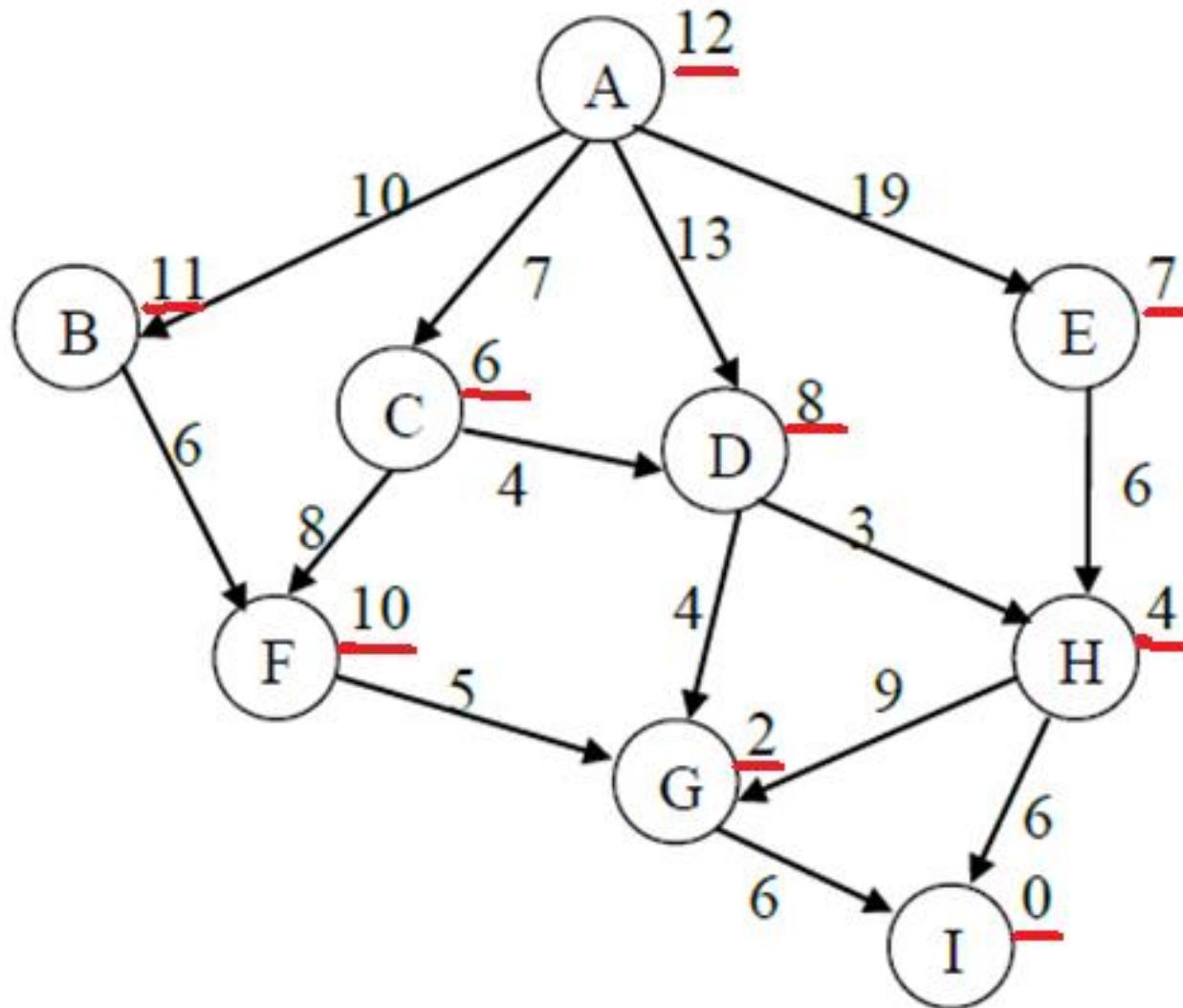
End;

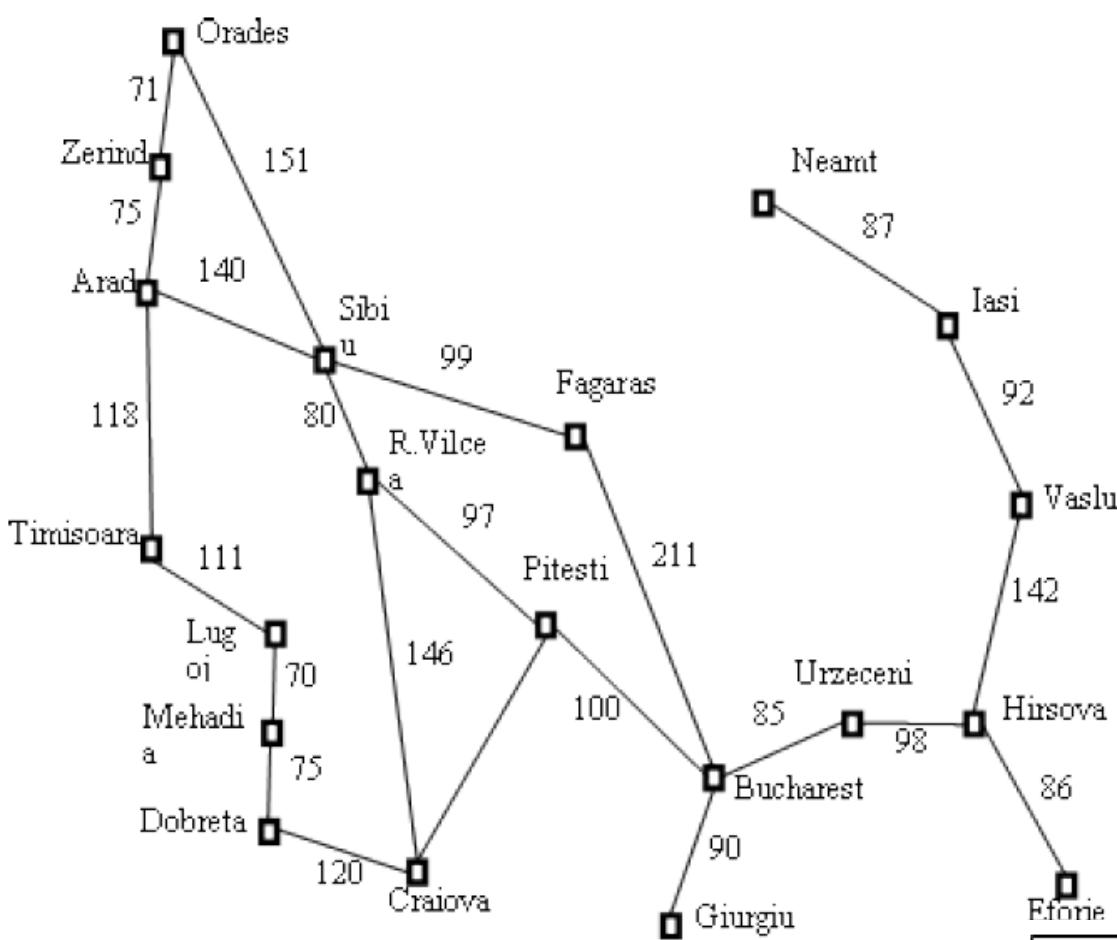


Thuật toán tìm kiếm nhánh và cành

TT	TTK	k(u,v)	h(v)	g(v)	f(v)	DS L1	Danh sách L
A	C	9	15	9	24	D13,E21,C 24,F27	D13,E21,C24,F27
	D	7	6	7	13		
	E	13	8	13	21		
	F	20	7	20	27		
D	H	8	10	15	25	E19,H25	E19,H25,E21,C24,F27
	E	4	8	11	19		
E	K	4	2	15	17	K17,I18	K17,I18,H25,E21,C24,F27
	I	3	4	14	18		
K	B	6	0	21	21	B21	B21,I18,H25,E21,C24,F27
B	TTKT,tìm được đường đi tạm thời, độ dài 21						I18,H25,E21,C24,F27
I	K	9	2	23	25	B19,K25	B19,K25,H25,E21,C24,F2
	B	5	0	19	19		7
B	TTKT,tìm được đường đi tạm thời, độ dài 19						K25,H25,E21,C24,F27

- Trạng thái đầu : A; TTKT: I





Arad	366	Mehadia	241
Bucharest	0	Neamt	234
Craiova	160	Oradea	380
Dobreta	242	Pitesti	98
Eforie	161	R. Vilcea	193
Fagaras	178	Sibiu	253
Giurgiu	77	Timisoara	329
Hirsova	151	Urziceni	80
Iasi	226	Vaslui	199
Lugoj	244	Zerind	374