CHƯƠNG 6: ĐỒ THỊ (GRAPH)

DATA STRUCTURES AND ALGORITHMS

L/O/G/O

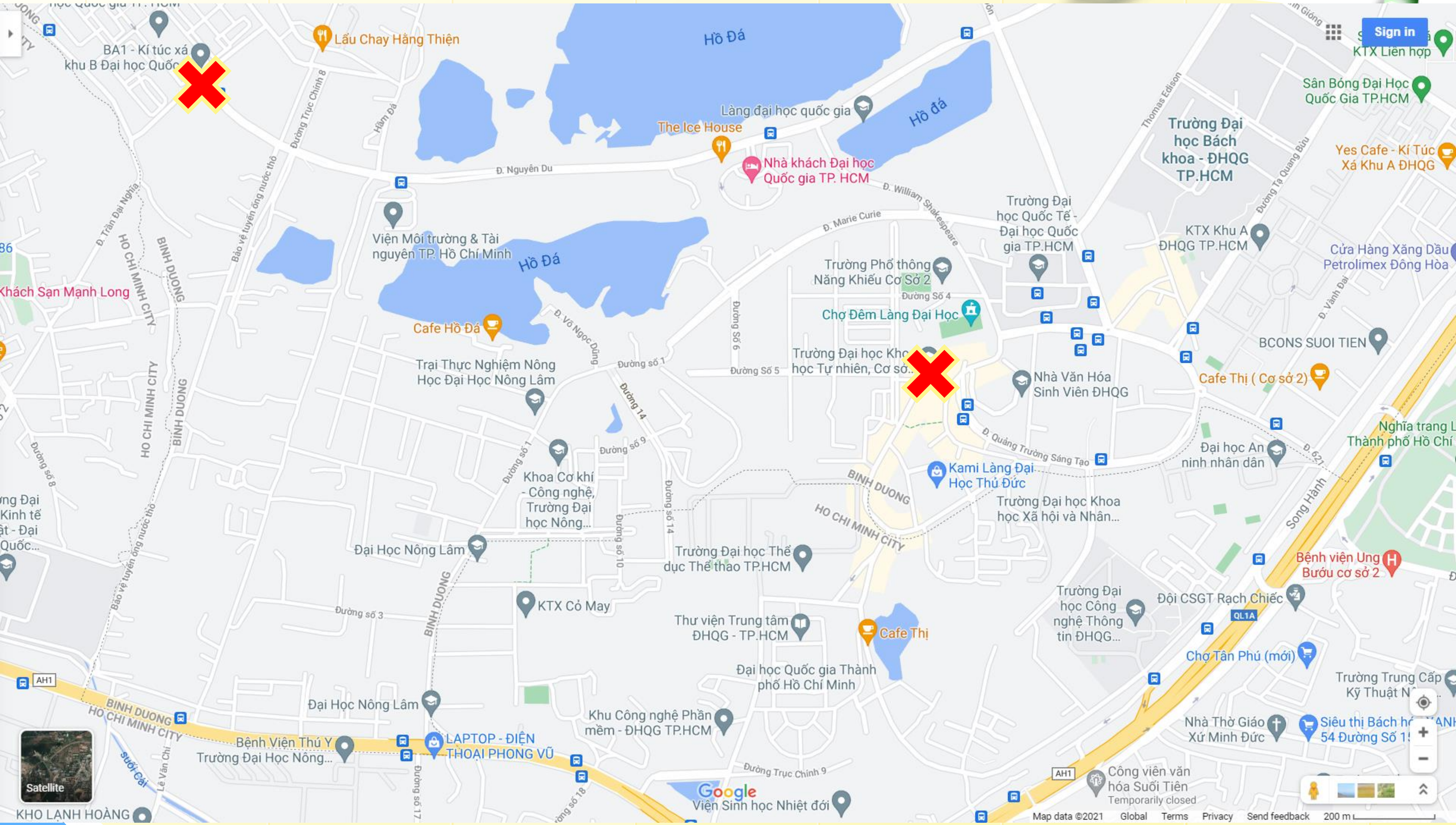
Nội dung buổi 3

6.1 Các khái niệm trên đồ thị

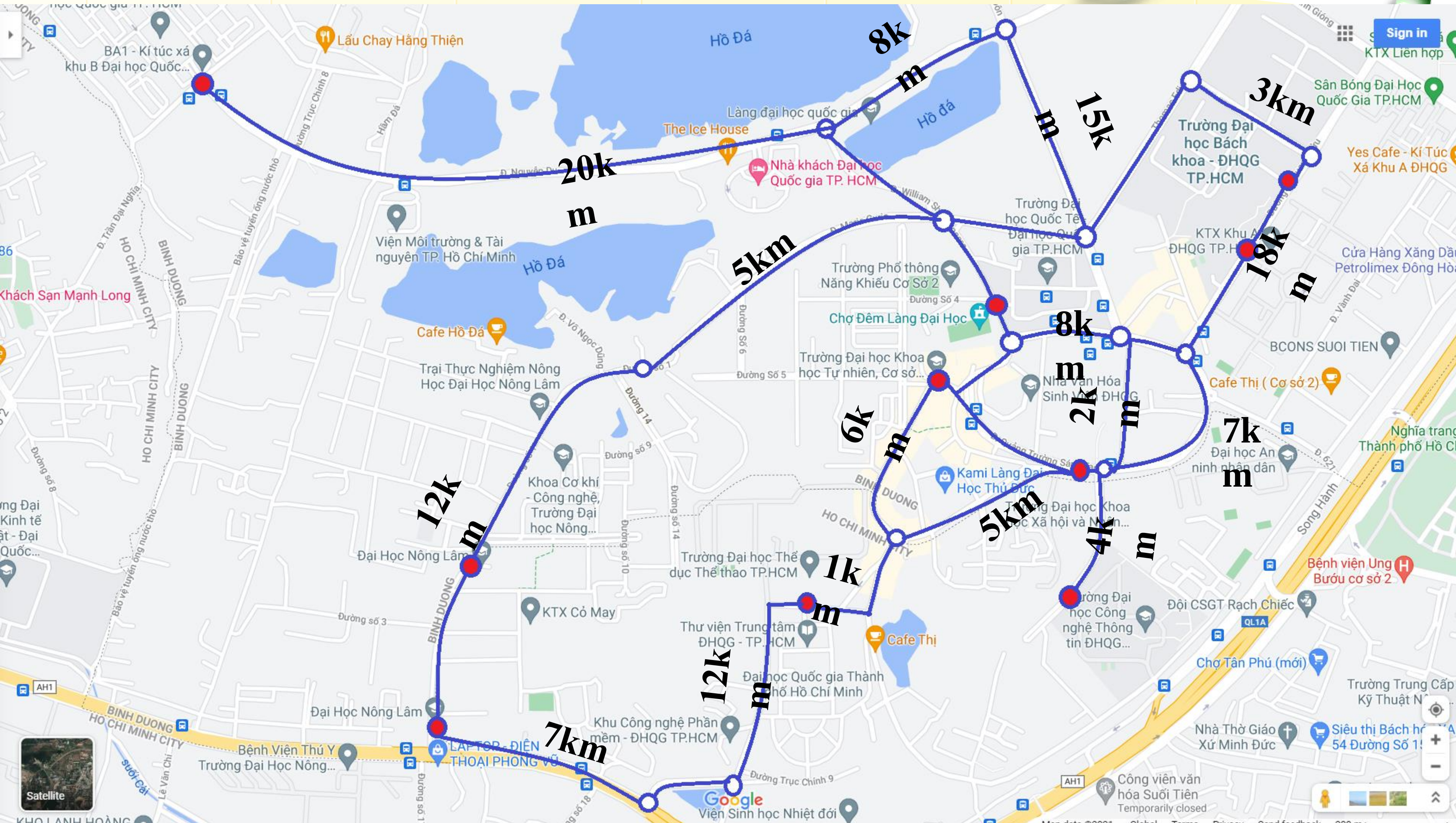
6.2 Biểu diễn đồ thị trên máy tính

6.3 Duyệt đồ thị

6.4 Thuật toán tìm đường đi ngắn nhất



Hồ Đá
Hồ đá
Hồ Đá
Lẩu Chay Hằng Thiện
BA1 - Kí túc xá khu B Đại học Quốc
Làng đại học quốc gia
The Ice House
Nhà khách Đại học Quốc gia TP. HCM
Viện Môi trường & Tài nguyên TP. Hồ Chí Minh
Đ. Nguyễn Du
Đ. Marie Curie
Đ. William Shakespeare
Khách Sạn Mạnh Long
Cafe Hồ Đá
Trại Thực Nghiệm Nông Học Đại Học Nông Lâm
Trường Phổ thông Năng Khiếu Cơ Sở 2
Chợ Đêm Làng Đại Học
Trường Đại học Khoa học Tự nhiên, Cơ sở...
Trường Đại học Quốc Tế - Đại học Quốc gia TP.HCM
Trường Đại học Bách khoa - ĐHQG TP.HCM
KTX Khu A ĐHQG TP.HCM
Sân Bóng Đại Học Quốc Gia TP.HCM
KTX Liên hợp
Yes Cafe - Kí Túc Xá Khu A ĐHQG
Cửa Hàng Xăng Dầu Petrolimex Đông Hòa
BCONS SUOI TIEN
Nhà Văn Hóa Sinh Viên ĐHQG
Cafe Thị (Cơ sở 2)
Nghĩa trang Thành phố Hồ Chí
Đại học An ninh nhân dân
Kami Làng Đại Học Thủ Đức
Trường Đại học Khoa học Xã hội và Nhân...
Đ. Quảng Trường Sáng Tạo
Khoa Cơ khí - Công nghệ, Trường Đại học Nông...
Đại Học Nông Lâm
Trường Đại học Thể dục Thể thao TP.HCM
KTX Cỏ May
Thư viện Trung tâm ĐHQG - TP.HCM
Cafe Thị
Trường Đại học Công nghệ Thông tin ĐHQG...
Đội CSGT Rạch Chiếc
Bệnh viện Ung Bướu cơ sở 2
Chợ Tân Phú (mới)
Đại học Quốc gia Thành phố Hồ Chí Minh
Khu Công nghệ Phần mềm - ĐHQG TP.HCM
Bệnh Viện Thú Y Trường Đại Học Nông...
LAPTOP - ĐIỆN THOẠI PHONG VŨ
Nhà Thờ Giáo Xứ Minh Đức
Siêu thị Bách hóa 54 Đường Số 1
Trường Trung Cấp Kỹ Thuật N
Công viên văn hóa Suối Tiên
Temporarily closed
Viện Sinh học Nhiệt đới
KHO LẠNH HOÀNG
BINH DUONG
HO CHI MINH CITY
Satellite
Sign in
Map data ©2021 Global Terms Privacy Send feedback 200 m



20km
8km
15km
3km
18km
5km
8km
2km
6km
7km
12km
5km
4km
1km
12km
7km
Lẩu Chay Hằng Thiện
Hồ Đá
BA1 - Kí túc xá khu B Đại học Quốc...
Làng đại học quốc gia
The Ice House
Hồ đá
Nhà khách Đại học Quốc gia TP. HCM
Sign in
KTX Liên hợp
Sân Bóng Đại Học Quốc Gia TP.HCM
Trường Đại học Bách khoa - ĐHQG TP.HCM
Yes Cafe - Kí Túc Xá Khu A ĐHQG
Viện Môi trường & Tài nguyên TP. Hồ Chí Minh
Hồ Đá
Khách Sạn Mạnh Long
Cafe Hồ Đá
Trường Phổ thông Năng Khiếu Cơ Sở 2
Chợ Đêm Làng Đại Học
Cửa Hàng Xăng Dầu Petrolimex Đông Hòa
BCONS SUOI TIEN
Trại Thực Nghiệm Nông Học Đại Học Nông Lâm
Trường Đại học Khoa học Tự nhiên, Cơ sở...
Cafe Thị (Cơ sở 2)
Đại học An ninh nhân dân
Nghĩa trang Thành phố Hồ Chí
Khoa Cơ khí - Công nghệ, Trường Đại học Nông...
Kami Làng Đại Học Thủ Đức
Đại Học Nông Lâm
Trường Đại học Thể dục Thể thao TP.HCM
Bệnh viện Ung Bướu cơ sở 2
Đội CSGT Rạch Chiếc
KTX Cỏ May
Thư viện Trung tâm ĐHQG - TP.HCM
Cafe Thị
Chợ Tân Phú (mới)
Đại học Quốc gia Thành phố Hồ Chí Minh
Trường Trung Cấp Kỹ Thuật
Đại Học Nông Lâm
Khu Công nghệ Phần mềm - ĐHQG TP.HCM
Nhà Thờ Giáo Xứ Minh Đức
Bệnh Viện Thú Y
Trường Đại Học Nông...
LAPTOP - ĐIỆN THOẠI PHONG VŨ
Công viên văn hóa Suối Tiên
Temporarily closed
Satellite
KHO LẠNH HOÀNG
Google
Viện Sinh học Nhiệt đới
Map data ©2021 Global Terms Privacy Send feedback 200 m

Các bài toán

1. Tìm đường đi ngắn nhất từ một đỉnh đến một đỉnh khác.
2. Tìm đường đi ngắn nhất từ một đỉnh đến các đỉnh còn lại
3. Tìm đường đi ngắn nhất giữa các cặp đỉnh

Thuật toán Dijkstra

❖ <u>Giới thiệu</u>:

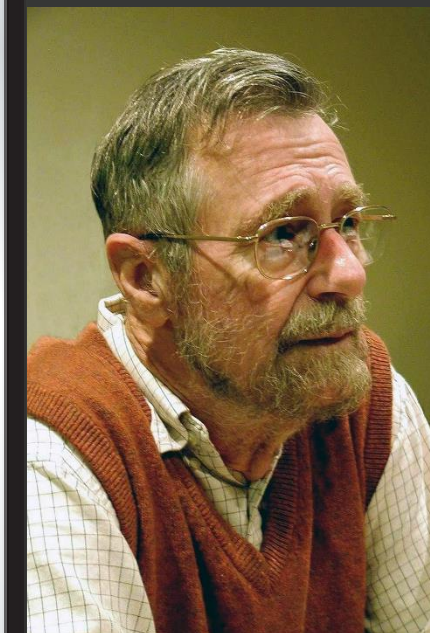
- Thuật toán tìm đường đi ngắn nhất từ một đỉnh xuất phát đến tất cả các đỉnh còn lại trong đồ thị có hướng, có trọng số không âm
- Được công bố lần đầu vào năm 1959
- Được đặt tên theo tên của nhà toán học và nhà vật lý người Hà Lan Edsger W. Dijkstra

Edsger Wybe Dijkstra

(/ˈdaɪkstrə/ *DYKE-strə*)

(11/5/1930 – 6/8/2002)

- Nhà toán học, nhà vật lý, nhà khoa học máy tính người Hà Lan
- Làm việc tại Trung tâm toán học, Viện nghiên cứu quốc gia về toán học và khoa học máy tính tại Amsterdam
- Giữ chức vị giáo sư tại Đại học Kỹ thuật Eindhoven, Hà Lan.
- Nghỉ hưu năm 2000



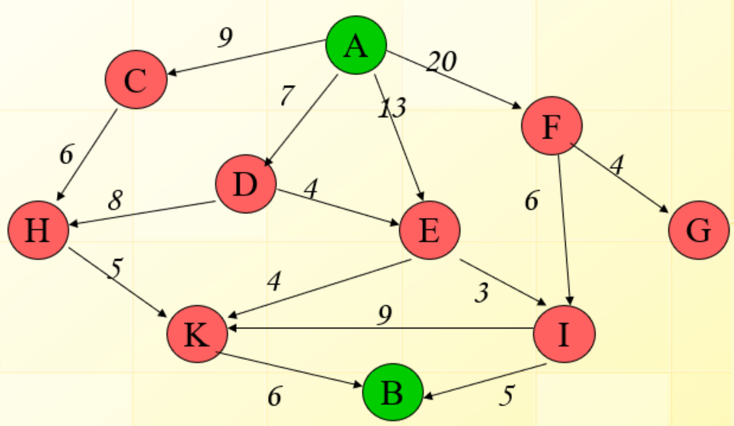
- Có các đóng góp mang tính chất nền tảng trong lĩnh vực ngôn ngữ lập trình
- Thuật toán Dijkstra, hệ điều hành THE và cấu trúc semaphore…

“Computer Science is no more about computers than astronomy is about telescopes”

Thuật toán Dijkstra

❖ Bài toán 1:

- **Input:** Đồ thị có hướng G=(V,E,w), w(i,j) ≥ 0 ∀ (i,j)∈E, đỉnh nguồn s, đỉnh đích g
- **Output:** đường đi ngắn nhất từ đỉnh s đến đỉnh g và độ dài của đường đi



Thuật toán Dijkstra

❖ <u>Ý tưởng chính</u>:

- Hàm d(u) dùng để lưu trữ độ dài đường đi (khoảng cách) ngắn nhất từ đỉnh nguồn s đến đỉnh u

$$d(u) = \min\{d(v) + w(v,u), v \in X(u)\}$$

$X(u)$ là tập tất cả các đỉnh có cạnh đi tới đỉnh u

- Đặt khoảng cách từ đỉnh nguồn s đến chính nó là 0 và đến tất cả các đỉnh khác là vô cùng
- Tiến hành lặp cho đến khi tất cả các đỉnh đã được xác định khoảng cách ngắn nhất từ s hoặc không còn đỉnh nào có thể đạt tới từ s
- Mỗi lần lặp, chọn đỉnh p chưa đi qua có giá trị d(p) nhỏ nhất, cập nhật khoảng cách của các đỉnh kề thông qua đỉnh được chọn p

Ký hiệu

Gọi Open : tập các đỉnh có thể xét ở bước kế tiếp, các đỉnh có thể được xem xét lại, đỉnh chờ duyệt

Close : tập các đỉnh đã xét/đã duyệt, không xem xét lại

s : đỉnh xuất phát

g : đỉnh đích

p : đỉnh đang xét, đỉnh hiện hành

$\Gamma(p)$: tập các đỉnh kề của p

$q \in \Gamma(p)$: một đỉnh kề của p

Dijkstra

❖ <u>Giải thuật tham khảo</u>:

Mỗi đỉnh p tương ứng với 1 số d(p): khoảng cách đi từ đỉnh ban đầu tới p

<u>Bước 1</u>: Khởi tạo

Open := {s};

Close :={};

d(s) :=0;

<u>Bước 2</u>: While (Open ≠ {})

2.1 Chọn p thuộc Open có d(p) nhỏ nhất

2.2 Nếu p là trạng thái kết thúc thì xuất đường đi, thoát

2.3 Nếu p đã duyệt rồi thì bỏ qua, trở lại đầu vòng lặp

2.4 Chuyển p qua Close, và **mở các đỉnh kế tiếp q sau p (kề với p)**

d(q) = d(p) + cost(p,q) ;

Thêm q vào Open

<u>Bước 3</u>: Không có kết quả.

Tìm kiếm Dijkstra

Bước 2: While (**Open ≠{}**)

2.4 Chuyển p qua Close, và mở các đỉnh kế tiếp q sau p

2.3.1 Nếu q đã có trong Open

nếu d(q)> d(p)+cost(p,q)

d(q) = d(p) + cost(p,q);

parent(q)=p ;

2.3.2 Nếu q chưa có trong Open và Close

d(q) = d(p) +cost(p,q);

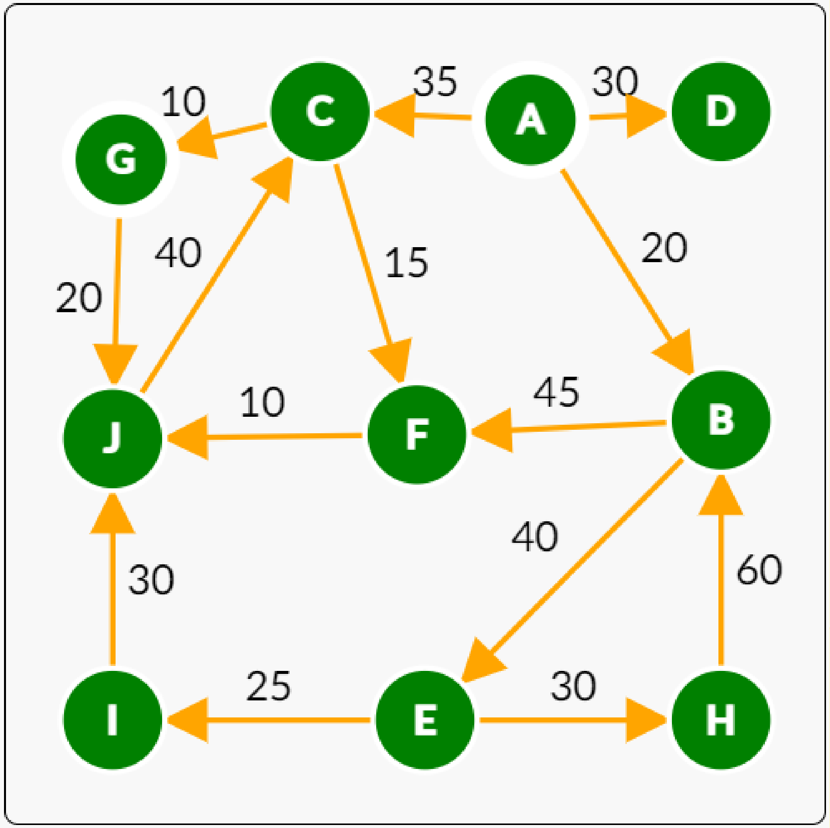
parent(q)=p;

Thêm q vào Open

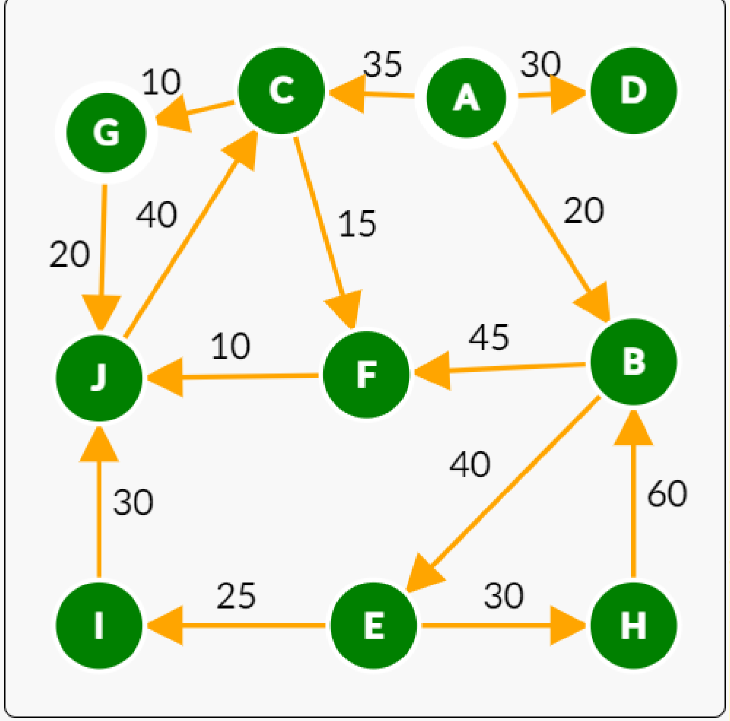
Câu hỏi thảo luận:

- Cách truy xuất đường đi?
- Nếu p hoặc q đã có trong Close thì có cần xử lý không?
- Độ phức tạp?

Ví dụ: Tìm đường đi ngắn nhất từ A đến J



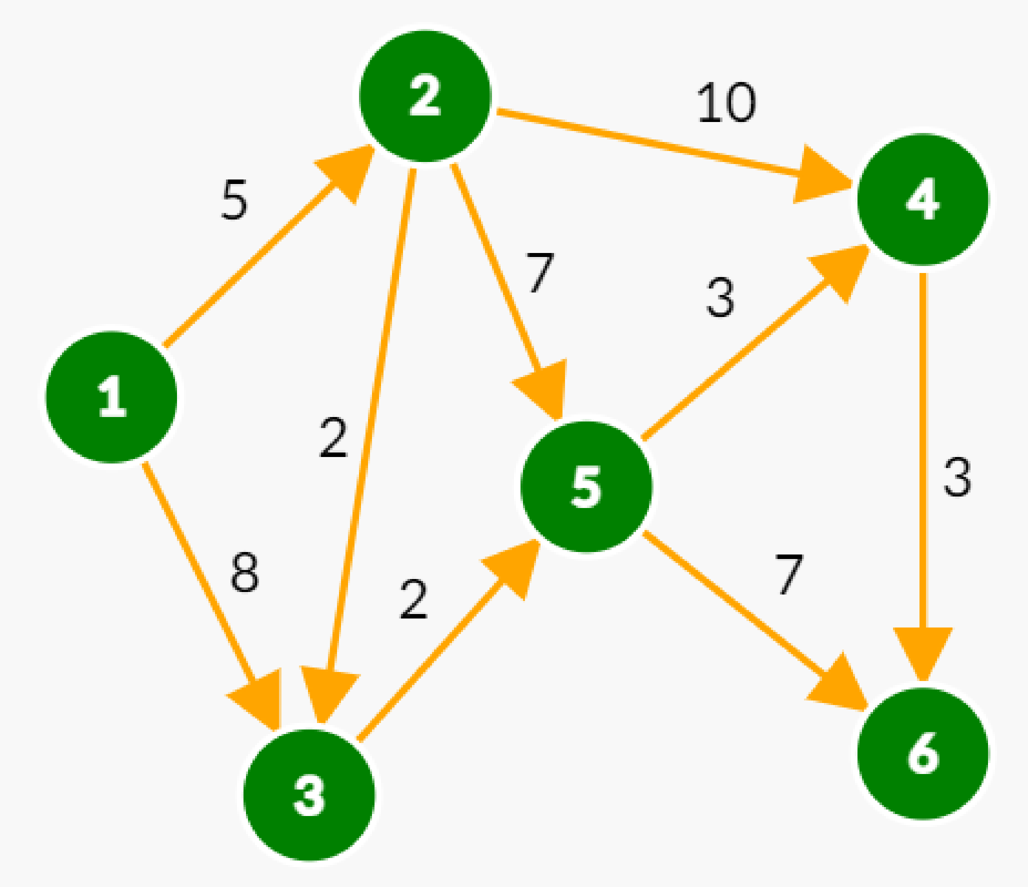

```
10 14
A B C D E F G H I J
A B 20
A C 35
A D 30
B E 40
B F 45
C F 15
C G 10
E H 30
E I 25
F J 10
G J 20
H B 60
I J 30
J C 40
```



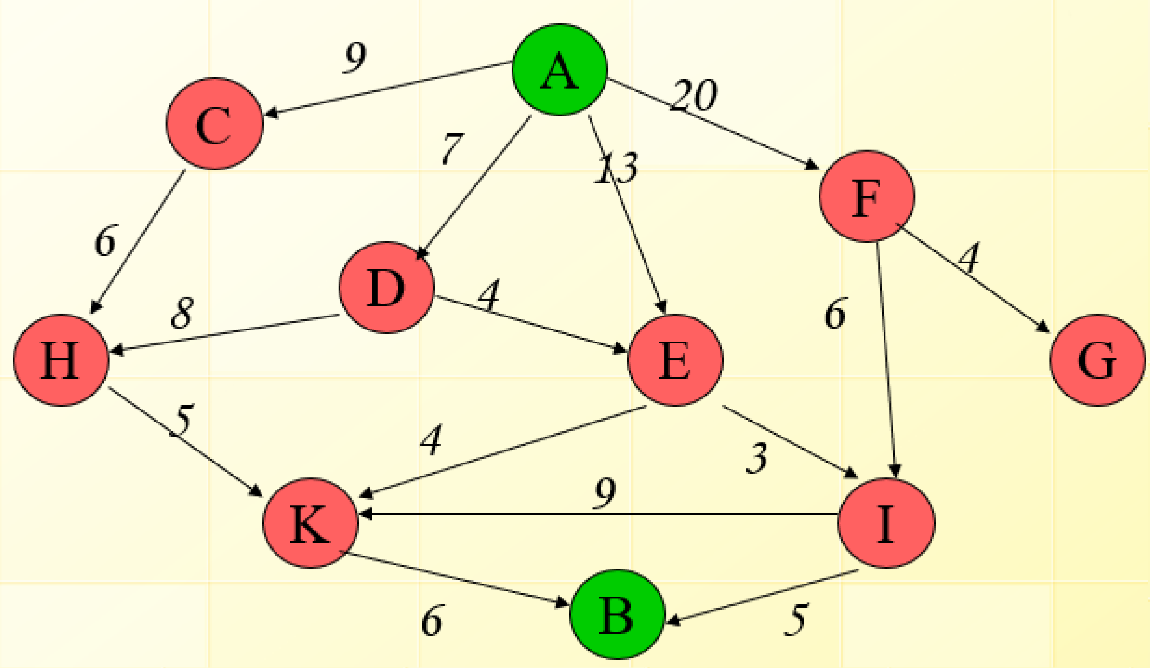
Ví dụ: Tìm đường đi ngắn nhất từ A đến J dùng Dijkstra

p	Γ(p)	Open	Close
		{(A,0)}	{}
A	{B,C,D}	{(B,20), (C,35), (D,30)}	{A}
B	{E,F}	{(C,35), (D,30), (E,60), (F,65)}	{A,B}
D	{}	{(C,35), (E,60), (F,65)}	{A,B,D}
C	{F,G}	{(E,60), ~~(F,65)~~, (F,50), (G,45)}	{A,B,D,C}
G	{J}	{(E,60), (F,50), (J,65)}	{A,B,D,C,G}
F	{J}	{(E,60), ~~(J,65)~~, (J,60)}	{A,B,D,C,G,F}
J			

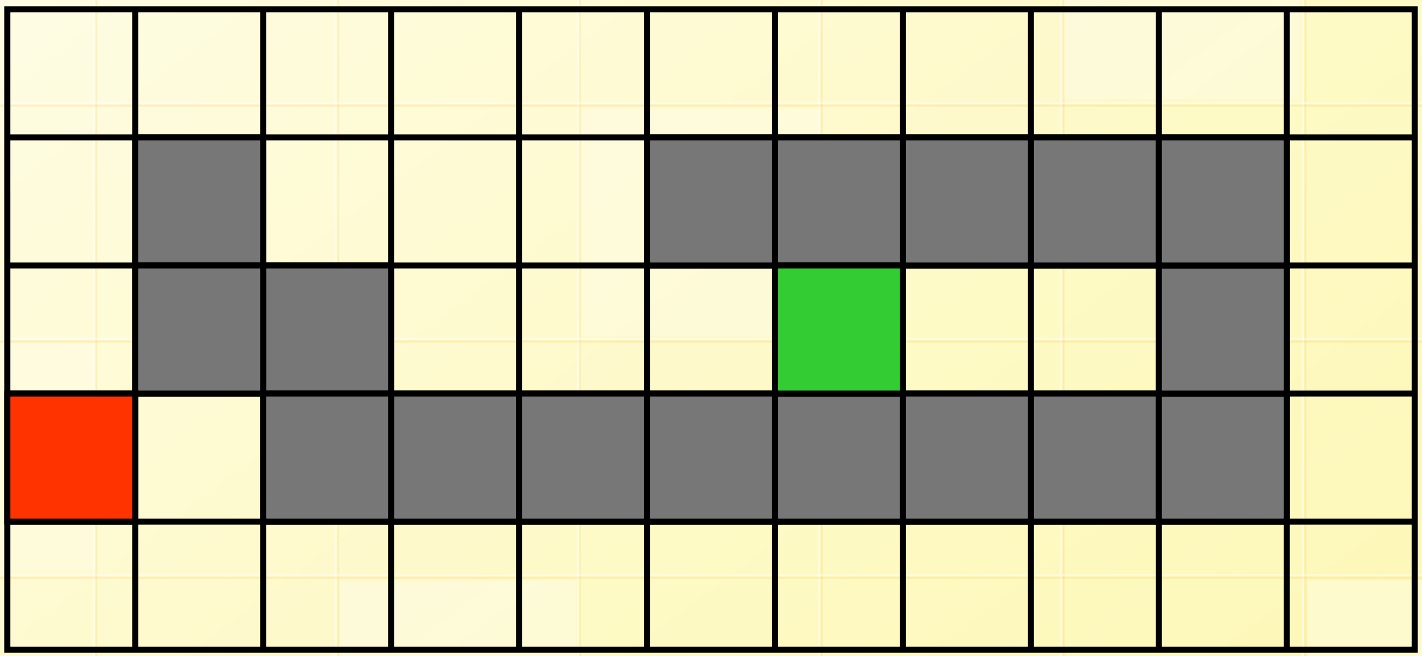
Bài tập tại lớp:
Tìm đường đi ngắn nhất từ 1 đến 6



Bài tập tại lớp:
Tìm đường đi ngắn nhất từ A đến B



Bài tập tại lớp: Robot tìm đường



Tìm kiếm Dijkstra

❖ Một số gợi ý về cài đặt:

1. Tổ chức cấu trúc dữ liệu

- vector<vector<int> > matrix; // ma trận trọng số của đồ thị
- vector<string> v_list; // lưu danh sách các tên đỉnh (chuỗi)
- map<string, int> v_index; // ánh xạ từ tên đỉnh sang index để có thể truy xuất thông tin trong ma trận kề
- priority_queue<….>open; // lưu các đỉnh chờ duyệt
- vector<bool> close(v, 0); // lưu thông tin đỉnh nào đã duyệt qua rồi
- map<string, string> parent; // lưu thông tin cha con, parent[u]=v nghĩa là cha của u và v
- vector<int> d(v,INF); // lưu khoảng cách (ngắn nhất) từ s đến các đỉnh khác

Priority Queue

Khởi tạo min heap từ priority queue

```
priority_queue< pair<int,int>, vector<pair<int,int>>,greater<pair<int,int>>> open;
```

Các hàm trên Priority Queue:

- size(): Trả về kích thước của hàng đợi
- empty(): kiểm tra có rỗng hay không
- top(): Lấy phần tử có độ ưu tiên cao nhất
- pop(): Xóa phần tử có độ ưu tiên cao nhất
- push(): Thêm 1 phần tử vào

Priority queue cũng giống như queue nhưng được thiết kế đặc biệt để phần tử ở đầu luôn luôn là phần tử có độ ưu tiên lớn nhất so với các phần tử khác

Tìm kiếm Dijkstra

❖ Một số gợi ý về cài đặt:

```
Bước 1: open.push({0,s}); d[s] = 0;
Bước 2: while (!open.empty())
            pair<int, int> top = open.top();  open.pop();
            int p=top.second, d=top.first;
            if (p == g) {found = true; break; }
            if(close[p]==1) continue;
    close[p] = 1;
            for (int i = 0; i < v; i++)
                if (matrix[p][i]>0 && close[i]== 0)
                    if (d[i]>d[p]+ matrix[p][i] )
                        d[i]=d[p]+ matrix[p][i] ; open.push({d[i],i});  parent[i] = p;
Bước 3: Xử lý xuất đường đi hoặc kết luận không tìm thấy
```

Tìm kiếm Dijkstra

- GV HƯỚNG DẪN SV CODE HOÀN CHỈNH THUẬT TOÁN TẠI LỚP