Đồ thị trọng số

Created by anhtt-fit@mail.hut.edu.vn
Edited by huonglt-fit@mail.hut.edu.vn

Đồ thị trọng số

- Có thể thêm thuộc tính vào cạnh, ta gọi đó là trọng số.
  - Ví dụ, nếu sử dụng đồ thị như bản đồ với các đỉnh là các thành phố, các cạnh là đường cao tốc nối các thành phố.
  - Tìm đường đi ngắn nhất giữa các thành phố
  - Tìm đường đi có chi phí nhỏ nhất giữa các thành phố

Đường đi ngắn nhất

- Đồ thị $G = (V,E)$ với hàm trọng số W: $E \rightarrow R$ (gán giá trị thực cho cạnh)
- Trọng số của đường đi $p = v_1 \rightarrow v_2 \rightarrow \ldots \rightarrow v_k$

$$w(p) = \sum_{i=1}^{k-1} w(v_i, v_{i+1})$$

- Đường đi ngắn nhất = đường đi với trọng số nhỏ nhất
- Ứng dụng
  - Định tuyến trên mạng tĩnh/động
  - Lập kế hoạch hoạt động của robot

Bài toán tìm đường đi ngắn nhất

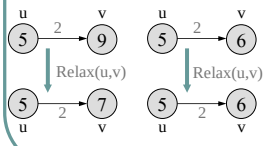
- Bài toán tìm đường đi ngắn nhất
  - **1 nguồn (1 đích).** Tìm đường đi ngắn nhất từ 1 nguồn cho trước đến đích
  - **1 cặp.** Cho 2 đỉnh, tìm đường đi ngắn nhất giữa chúng.
  - **Tất cả các cặp.** Tìm các đường đi ngắn nhất giữa các cặp đỉnh. Lập trình động.

Trọng số âm hay chu trình?

- Trọng số âm chấp nhận khi không có chu trình có trọng số âm (nếu không thì có thể thêm chu trình có trọng số âm vào đường đi ngắn nhất)
- Đường đi ngắn nhất có thể không có chu trình (nếu không thì ta có thể cải thiện bằng cách loại bỏ chu trình)
  - Bất cứ đường đi nào trên đồ thị G đều có <= n – 1 cạnh, với n là số đỉnh

Relaxation

- Với mỗi đỉnh v trên đồ thị, ta có 1 hàm giá *v*.**d**() để quản lý chi phí đường đi ngắn nhất từ s, khởi đầu = ∞
- Relaxing cạnh(*u,v*) nghĩa là kiểm tra xem có thể cải thiện đường đi ngắn nhất đến v không bằng cách đi qua u.




```
Relax (u,v,G)
if v.d() > u.d()+G.w(u,v)
  then
   v.setd(u.d()+G.w(u,v))
   v.setparent(u)
```

Thuật toán Dijkstra

- Không có cạnh có trọng số âm
- Giống tìm kiếm rộng (nếu tất cả các trọng số = 1, dùng BFS)
- Dùng Q, 1 hàng đợi ADT được quyết định bởi hàm *v*.**d**() (BFS sử dụng hàng đợi FIFO, ở đây ta dùng PQ, được tổ chức lại với 1 khoảng cách d nào đó)
- Ý tưởng
  - Duy trì tập S các đỉnh đã thăm
  - Tại mỗi bước, chọn đỉnh gần nhất u, thêm vào S, giải phóng tất cả các cạnh từ u

Demo

- demo-dijkstra.ppt

Dijkstra's Pseudo Code

- Input: Graph *G*, start vertex *s*

```
Dijkstra(G,s)
01 for each vertex u ∈ G.V()
02    u.setd(∞)
03    u.setparent(NIL)
04 s.setd(0)
05 // Set S is used to explain the algorithm
06 Q.init(G.V())  // Q is a priority queue ADT
07 while not Q.isEmpty()
08    u ← Q.extractMin()
09
10    for each v ∈ u.adjacent() do
11       Relax(u, v, G)
12       Q.modifyKey(v)
```

relaxing edges

Cài đặt

- Biến đổi graph API cho phép đưa vào trọng số cạnh như sau:

```
#define INFINITIVE_VALUE  10000000
typedef struct {
  JRB edges;
  JRB vertices;
} Graph;
void addEdge(Graph graph, int v1, int v2, double weight);
double getEdgeValue(Graph graph, int v1, int v2); // return
   INFINITIVE_VALUE nếu không có cạnh giữa v1 và v2
int indegree(Graph graph, int v, int* output);
int outdegree(Graph graph, int v, int* output);
double shortedPath(Graph graph, int s, int t, int* path,
   int*length); // return tổng giá trị đường đi, đường đi. Return
   INFINITIVE_VALUE nếu không tìm thấy đường đi
```

Bài tập

- Viết chương trình cài đặt đồ thị trọng số dùng API. Thử nghiệm với ví dụ sau:

```
Graph g = createGraph();
// thêm đỉnh và cạnh vào đồ thị
int s, t, length, path[1000];
double weight = shortedPath(g, s, t, path, &length);
if (weight == INFINITIVE_VALUE)
  printf("No path between %d and %d\n", s, t);
else {
  printf("Path between %d and %d:", s, t);
  for (i=0; i<length; i++) printf("%4d", path[i]);
  printf("Total weight: %f", weight);
}
```

Bài tập lớn II

- Mục đích của bài này là mô phỏng bản đồ xe buýt Hà Nội
- Trước tiên, bạn cần sưu tập dữ liệu về hệ thống các tuyến buýt Hà nội theo dạng đồ thị, trong đó:
  - Mỗi đỉnh là 1 trạm xe buýt ứng với 1 điểm trong Hà nội
  - Mỗi cạnh nối trạm xe buýt thông qua tuyến buýt.
  - Ví dụ, có 16 trạm trên tuyến buýt 1A: "Yên Phụ - Hàng Đậu - Hàng Cót - Hàng Gà - Hàng Điếu - Đường Thành - Phủ Doãn - Triệu Quốc Đạt - Hai Bà Trưng - Lê Duẩn - Khâm Thiên - Nguyễn Lương Bằng- Tây Sơn - Nguyễn Trãi - Trần Phú (Hà Đông) - Bến xe Hà Đông"
  - Tham khảo tại http://www.hanoibus.com.vn/InfobusVN/hanoibus/index.asp?pPage=lotrinh.htm

Bài tập lớn II

- Mỗi cạnh trên đồ thị được gắn với tuyến bus từ 1 điểm này đến điểm khác. Ví dụ, cạnh “Yên Phụ - Trần Nhật Duật” được gắn với 4A, 10A.
- Tổ chức và lưu trữ dữ liệu trong 1 file nạp vào chương trình khi chạy.
- Viết lại graph API để lưu bản đồ bus trong bộ nhớ
- Xây dựng hàm để tìm đường đi ngắn nhất giữa 2 điểm. Ví dụ từ “Yên Phụ” đến “Ngô Quyền”.