

BÀI TẬP THỰC HÀNH TÌM ĐƯỜNG ĐI NGẮN NHẤT – DIJKSTRA

1 Qui định

- Hạn nộp: xem trên Moodle
- Loại bài tập: **cá nhân**
- Nén toàn bộ thư mục thành tập tin <MSSV>.zip hoặc <MSSV>.rar.

Lưu ý: tất cả các bài làm sai qui định sẽ không được chấm (tức 0 điểm)

2 Nội dung

2.1 Yêu cầu bắt buộc

Viết chương trình tìm đường đỉnh ngắn nhất giữa hai đỉnh cho trước của đồ thị **G** bằng thuật toán **Dijkstra**. Biết rằng **G** là đồ thị đơn, có hướng và các cạnh có trọng số dương. Chương trình chạy dạng tham số dòng lệnh như sau:

<Tên chương trình> <tập tin đầu vào> <tập tin đầu ra> <đỉnh bắt đầu> <đỉnh kết thúc>

Tập tin đầu vào chương trình có định dạng như sau:

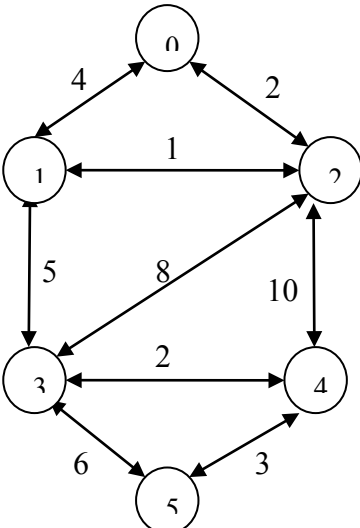
- **Dòng đầu tiên:** số đỉnh của đồ thị (gọi là N).
- **N dòng tiếp theo:** ma trận kề của đồ thị với qui ước
 - $A[i][j] > 0$: trọng số của cạnh nối trực tiếp từ đỉnh i đến đỉnh j.
 - $A[i][j] = 0$: không có cạnh nối trực tiếp từ đỉnh i đến đỉnh j.

Tập tin đầu ra của chương trình có nội dung như sau:

- Nếu không tồn tại đường đi ngắn nhất giữa hai đỉnh s và g thì xuất là **-1**.
- Ngược lại, thì xuất những thông tin theo định dạng sau:
 - **Dòng đầu tiên:** chứa chi phí để đi từ s đến g (tổng trọng số các cạnh phải đi qua).
 - **Dòng thứ 2:** đường đi ngắn nhất từ s đến g. Chính là danh sách các đỉnh lần lượt phải đi qua

Trong trường hợp tồn tại nhiều hơn một đường đi ngắn nhất từ s đến g thì chỉ cần xuất **một đường đi** tùy ý trong số đó.

Ví dụ: tìm đường đi ngắn nhất từ đỉnh 0 đến đỉnh 5.

Tập tin đầu vào	Đồ thị	Tập tin đầu ra
6 0 4 2 0 0 0 4 0 1 5 0 0 2 1 0 8 10 0 0 5 8 0 2 6 0 0 10 2 0 3 0 0 0 6 3 0		13 0 2 1 3 4 5

2.2 Yêu cầu nâng cao(cộng điểm):

a. Mở rộng 1 : Hãy mở rộng thuật toán Dijkstra để có thể tìm được đường đi ngắn nhất giữa đỉnh s cho trước với tất cả các đỉnh khác của đồ thị. Trong trường hợp, tồn tại nhiều hơn một đường đi ngắn nhất giữa 2 đỉnh thì chỉ xuất một đường đi trong số đó. Chương trình chạy với tham số dòng lệnh như sau:

<Tên chương trình> <Tập tin đầu vào> <Tập tin đầu ra> <Đỉnh bắt đầu>

Tập tin đầu vào sẽ có định dạng giống như trong yêu cầu cơ bản, riêng dòng cuối cùng chỉ ghi chỉ số của đỉnh s .

Tập tin đầu ra sẽ liệt kê thông tin về các đường đi ngắn nhất giữa đỉnh s với các đỉnh còn lại. Tương ứng với mỗi cặp đỉnh (s, v) (với v khác s), nội dung xuất ra sẽ là:

- Nếu tồn tại đường đi:
 - Dòng thứ nhất:** <chỉ số của đỉnh s > <chỉ số của đỉnh v >
 - Dòng thứ hai:** chi phí của đường đi ngắn nhất từ s đến v
 - Dòng thứ ba:** danh sách các đỉnh phải đi từ s đến v .
- Nếu không tồn tại đường đi:

- **Dòng thứ nhất:** <chỉ số của đỉnh s> <chỉ số của đỉnh v>
- **Dòng thứ hai:** -1 (nghĩa là không tồn tại đường đi).

Nếu đồ thị có N đỉnh, thì tập tin xuất sẽ có N-1 đoạn nội dung có định dạng như trên.

Ví dụ: Tìm đường đi ngắn nhất từ đỉnh 0 đến các đỉnh còn lại.

Tập tin đầu vào	Đồ thị	Tập tin đầu ra
6 0 4 2 0 0 0 4 0 1 5 0 0 2 1 0 8 10 0 0 5 8 0 2 6 0 0 10 2 0 3 0 0 0 6 3 0		0 1 3 0 2 1 0 2 2 0 2 0 3 8 0 2 1 3 0 4 10 0 2 1 3 4 0 5 13 0 2 1 3 4 5

b. Mở rộng 2: Hãy mở rộng thuật toán Dijkstra để có thể tìm tất cả các đường đi ngắn nhất từ s đến g. Tham số dòng lệnh tương tự yêu cầu cơ bản. Tập tin đầu ra sẽ có định dạng như sau:

- **Dòng đầu tiên:** chi phí của đường đi ngắn nhất từ s đến g.
- **Các dòng tiếp theo:** mỗi dòng ghi một đường đi ngắn nhất từ s đến g. Thứ tự ghi phải đảm bảo:

- Số đỉnh của đường đi ở một dòng luôn lớn hơn hoặc bằng số đỉnh của đường đi ở dòng liền trước và luôn nhỏ hơn hoặc bằng số đỉnh của dòng liền sau.
- Nếu hai đường đi **s1** và **s2** có số đỉnh bằng nhau thì s1 sẽ ghi trước s2 nếu chỉ số đỉnh thứ k bất kỳ của s1 luôn luôn nhỏ hơn hoặc bằng chỉ số đỉnh thứ k của s2.

Ví dụ:

<i>Tập tin đầu vào</i>	<i>Đồ thị</i>	<i>Tập tin đầu ra</i>
9 0 2 1 0 0 0 0 0 10 0 0 0 3 0 0 0 0 0 0 1 0 4 0 0 0 0 9 0 0 0 0 1 3 0 0 5 0 0 0 0 0 1 1 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 2 0 0 0 0 0 0 0 0 0		10 0 8 0 2 8 0 1 3 8 0 2 3 8 0 2 1 3 8 0 1 3 4 6 7 8 0 2 3 4 6 7 8 0 2 1 3 4 6 7 8