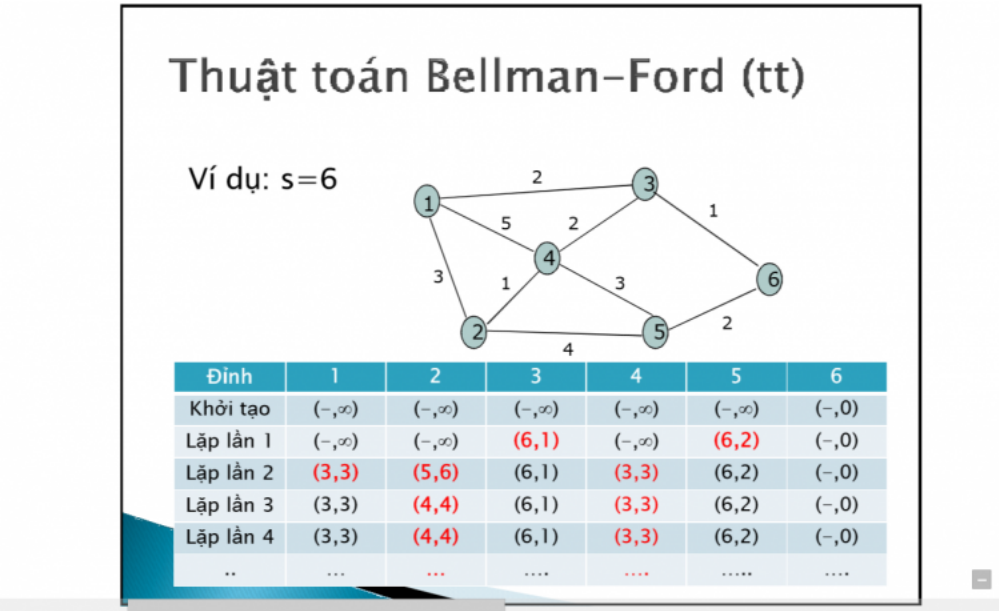
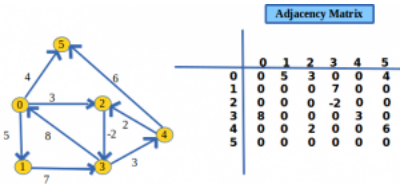
nguyenlephong1997@gmail.com 01643081355



Thuật toán bellman ford -Tìm đường đi ngắn nhất

23 Th12,2017 adminlephong Để lại bình luận

0 SHARES



Thuật toán bellman ford-Thuật toán tìm đường đi ngắn nhất

Xin chào tất cả các bạn,Ở các bài trước các bạn đã biết được các thuật toán như Dijkstra , Floy, Và bài hôm này Phong sẽ giới thiệu cho các bạn thuật toán tìm đường đi ngắn nhất khác, đó là thuật toán bellman ford.Tương tự như các bài khác ta lướt sơ qua một số lý thuyết nhé.hii.

Thuật toán Bellman-Ford là một thuật toán tính các đường đi ngắn nhất nguồn đơn trong một đồ thị có hướng có trọng số (trong đó một số cung có thể có trọng số âm). Thuật toán Dijsktra giải cùng bài toán này tuy nhiên Dijsktra có thời gian chạy nhanh hơn đơn giản là đòi hỏi trọng số của các cung phải có giá trị không âm.

BÀI VIẾT GẦN ĐÂY

Tài liệu các môn học của ngành công ng thông tin-Đại học Nông Lâm TP HCM

Source code bài tập lý thuyết đồ thị jav thi tham khảo

Thuật toán bellman ford -Tìm đường đ nhất

Hệ thống quản lý xe và bán vé xe onlin

Thuật toán floyd – Tìm đường đi ngắn các đỉnh trong đồ thị

Theo bạn "bạn thân" là gì nè?

Bằng cách nào để load dữ liệu từ các fi nhau lên đối tượng để xử lý ?

Duyệt cây trong đồ thị dùng thuật toár BFS

Tìm cây bao trùm nhỏ nhất dùng thuật kruskal

Tìm cây bao trùm nhỏ nhất dùng thuật prim

HỌC TẬP

Chọn chuyên mục

THỐNG KÊ

Online Users: 1

Today's Visits: 85

Last 30 Days Visits: 1.329

Total Visitors: 376

Total Posts: 14

Total Comments: 0

Thuật toán Bellman Ford chạy trong thời gian $O(V \cdot E)$, trong đó V là số đỉnh và E là số cung của đồ thị.

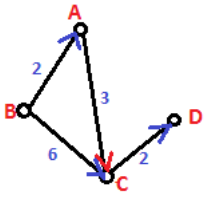
Ưu điểm:

– Từ 1 đỉnh xuất phát nhìn hình ta có thể suy ra đường đi ngắn nhất từ đỉnh đó tới các đỉnh khác mà không cần làm lại từ đầu.

– Ví dụ: Từ đỉnh 1 ta có thể tìm đường đi ngắn nhất từ 1->3 và 1->4 mà không cần làm lại.

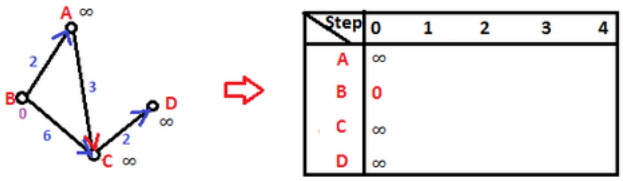
Chúng ta tham khảo ví dụ của wiki để hiểu thêm xíu nữa nhé!

Tìm đường đi ngắn nhất từ đỉnh B tới đỉnh D của đồ thị G



Đồ thị G

– **Bước 0:** Ta đánh dấu đỉnh xuất phát = 0, các đỉnh còn lại bằng vô cực.

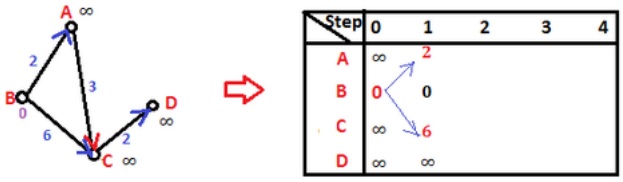


Bước 0

– **Bước 1:**

Tại đỉnh A có đỉnh B đi vào có chi phí hiện tại (2) < chi phí trước (∞) => cập nhật lại chi phí đỉnh A

Tại đỉnh C có đỉnh B đi vào có chi phí hiện tại (6) < chi phí trước (∞) => cập nhật lại chi phí đỉnh C



Bước 1

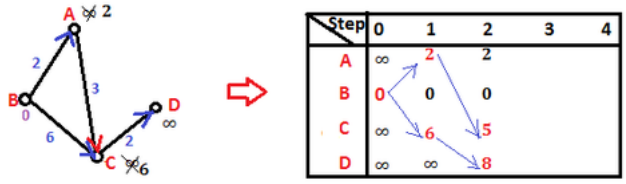
– **Bước 2:**

Tại đỉnh C có đỉnh A đi vào có chi phí hiện tại (5) < chi phí trước (6) => cập nhật lại chi phí đỉnh C

CÁC THẺ LIÊN KẾT

ABSTRACT CLASS (1) BELLMAN FORD (1) BFS (1) BẠN THÂN (1) CHU TRÌNH (1) CHUYEN HOM NAY (2) DFS (1) DIJKSTRA (1) DUYỆT CÂY (1) FLOYD (1 HAMILTON (1) JAVA (3) KRUSKA LOAD DỮ LIỆU (1) LẬP TRÌNH MẠNG (1) LÝ THUYẾT ĐỒ THỊ OOAD (1) PRIM (1) PROJECT (1) SOURCE CODE (1) THAO TÁC CẠNH TRONG ĐỒ THỊ (1) THIẾT K THUẬT TOÁN (1) TÌM CÂY BAO TRÙ TÌM ĐƯỜNG ĐI NGẮN NHẤT (ĐỀ THI (1) ĐƯỜNG ĐI (1)

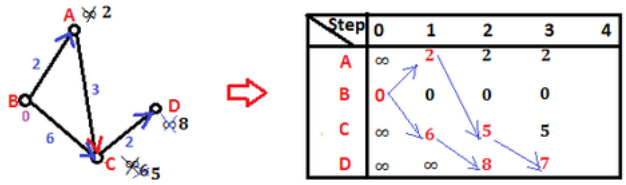
Tại đỉnh D có đỉnh C đi vào có chi phí hiện tại (8) < chi phí trước (∞) => cập nhật lại chi phí đỉnh D



Bước 2

– **Bước 3:**

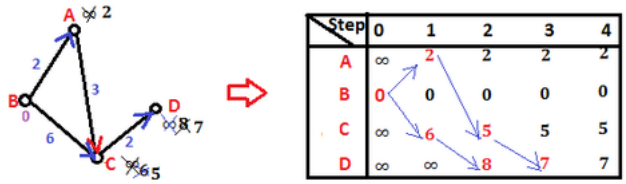
Tại đỉnh D có đỉnh C đi vào có chi phí hiện tại (7) < chi phí trước (8) => cập nhật lại chi phí đỉnh D



Bước 3

– **Bước 4:**

Bước 4 giống bước 3 nên thuật toán dừng.

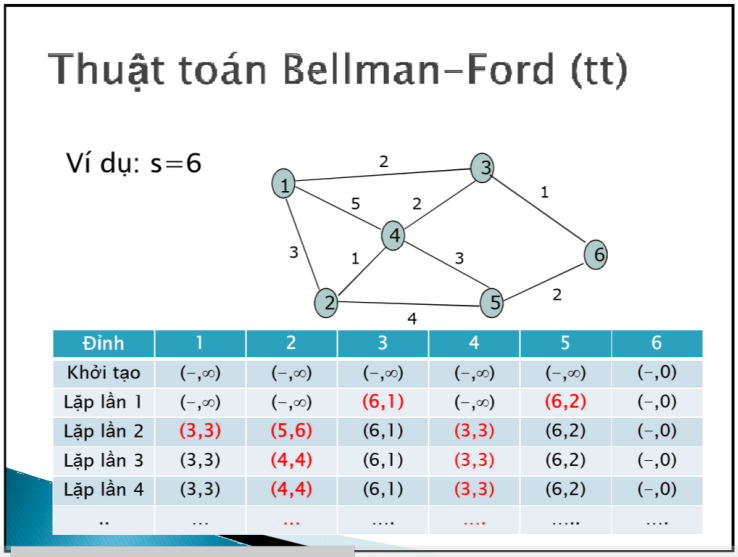


Bước 4

– **Kết luận:**Có đường đi ngắn nhất từ B->D: B->A->C->D

– **Lưu ý:**Nếu Bước 4 không giống bước 3 => kết luận không có đường đi ngắn nhất từ B->D

Một ví dụ khác qua hình ảnh dưới đây:



Đỉnh	1	2	3	4	5	6
Khởi tạo	(–,∞)	(–,∞)	(–,∞)	(–,∞)	(–,∞)	(–,0)
Lặp lần 1	(–,∞)	(–,∞)	(6,1)	(–,∞)	(6,2)	(–,0)
Lặp lần 2	(3,3)	(5,6)	(6,1)	(3,3)	(6,2)	(–,0)
Lặp lần 3	(3,3)	(4,4)	(6,1)	(3,3)	(6,2)	(–,0)
Lặp lần 4	(3,3)	(4,4)	(6,1)	(3,3)	(6,2)	(–,0)
..	…	…	….	….	…..	….

Đọc tới đây các bạn đã hình dung được thuật toán này chạy như thế nào chưa nè,chúng ta đi qua mã giả của nó nhé!

```
function BellmanFord(danh_sách_đỉnh, danh_sách_cung, nguồn)
  // hàm yêu cầu đồ thị đưa vào dưới dạng một danh sách đỉnh, một danh sách cung
  // hàm tính các giá trị khoảng_cách và đỉnh_liền_trước của các đỉnh,
  // sao cho các giá trị đỉnh_liền_trước sẽ lưu lại các đường đi ngắn nhất.

  // bước 1: khởi tạo đồ thị
  for each v in danh_sách_đỉnh:
      if v is nguồn then khoảng_cách(v):= 0
      else khoảng_cách(v):= vô cùng
      đỉnh_liền_trước(v):= null

  // bước 2: kết nạp cạnh
  for i from 1 to size(danh_sách_đỉnh)-1:
      for each (u,v) in danh_sách_cung:
          if khoảng_cách(v) > khoảng_cách(u) + trọng_số(u,v):
              khoảng_cách(v):= khoảng_cách(u) + trọng_số(u,v)
              đỉnh_liền_trước(v):= u

  // bước 3: kiểm tra chu trình âm
  for each (u,v) in danh_sách_cung:
      if khoảng_cách(v) > khoảng_cách(u) + trọng_số(u,v):
          error "Đồ thị chứa chu trình âm"
```

Đọc đoạn mã giả ở trên xong là hông biết gì luôn phải hông nè,hii.

Đoạn mã đó chúng ta chỉ cần tập trung vào cái if là ok.Câu if này thể hiện nếu có cạnh nó sẽ dán nhãn đỉnh nó đi qua cũng với trọng số.Việc này lặp đi lặp lại n lần thì sẽ hoàn thành .Nó tương tự như dijsktra và nhanh như floy phải hông nè.Nó khá giống với dijsktra phải hông nè.hii.

Các bạn tham khảo đoạn code dưới đây xem nó hoạt động như thế nào nhé!

```
    public class Graph{
...
    //Ví dụ này làm giống như hình ở vi dụ thứ 2 ở trên
    private int maxValue = 1000;
    //Mãng lưu đường đi P
    public int P[];
    //Mãng để cập nhật nhãn
    public int L[];
    //Tim duong di ngan nhat dung thuat toan bell man ford
    public void bellmanFord(int source) {
        // Khoi tao
        int n = topNum();
        // Vì mãng của Phong nhận vào load từ file nên chô nào vô cùng là Phong để s
        for (int i = 0; i < matrixA.length; i++) {
            for (int j = 0; j < matrixA.length; j++) {
                if (matrixA[i][j] == 0)
                    matrixA[i][j] = maxValue;
            }
        }
```

Copy

Ambiance

Mình chúc các bạn hoàn thành xong thuật toán này nhé,chúc các bạn có mùa giáng sinh vui vẻ, hạnh phúc, ấm áp bên gia đình và người thân nhé!

À đừng quên là nếu có thắc mắc gì thì đừng ngại comment nhé bạn thân! hii.

Hệ thống quản lý xe và bán vé xe online

Source code bài tập lý thuyết đồ thị java – Đề thi tham khảo

Leave a Reply

THEO DÕI TÔI BẠN NHÉ!

BÀI VIẾT PHỔ BIẾN

Thiết kế abstract class cho đồ thị

Tháng Mười Hai 4th, 2017

Cách tìm đường đi và chu trình hamilton bằng ngôn ngữ java

Tháng Mười Hai 4th, 2017

Tìm đường đi ngắn nhất dùng thuật toán dijkstra

Tháng Mười Hai 4th, 2017

BLOG IT NLU