**HỌC VIỆN KỸ THUẬT QUÂN SỰ**

**BÁO CÁO**

**Bài tập kết thúc môn**

**Phân tích và thiết kế giải thuật**

**Sinh viên: Đặng Văn Đức**

**Thuật toán Dijkstra**

1. **Nêu bài toán**

Thuật toán Dijkstra là một thuật toán giải quyết bài toán đường đi ngắn nhất nguồn đơn trong một đồ thị có hướng không có cạnh mang trọng số âm. Thuật toán thường được sử dụng trong định tuyến với một chương trình con trong các thuật toán đồ thị hay trong công nghệ hệ thống định vị toàn cầu (GPS).

Cho một đồ thị có hướng G = (V,E), một hàm trọng số w. E không âm và một đỉnh nguồn s. cần tính toán được đường đi ngắn nhất từ đỉnh nguồn s đến mỗi đỉnh của đồ thị.

Ví dụ: chúng ta dùng các đỉnh của đồ thị để mô hình các thành phố và các cạnh để mô hành các đường nối giữa chúng. Khi đó trọng số các cạnh có thể xem như độ dài của các con đường ( và do đó là không âm ). Chúng ta cần vận chuyển từ thành phố s đến thành phố t. Thuật toán Dijkstra sẽ giúp chỉ ra đường đi ngắn nhất chúng ta có thể đi.

Trọng số không âm của các cạnh của đồ thị mang tính tổng quát hơn khoảng cách hình học giữa 2 đỉnh đầu mút của chúng. Ví dụ, với 3 đỉnh A,B,C đường đi A-B-C có thể ngắn hơn so với đường đi trực tiếp A-C.

1. **Mô tả chi tiết thuật toán.**

Ý tưởng (tham lam): Có đồ thị G=(V,E), s0.

– L(v): độ dài đường đi ngắn nhất từ s0 đến đỉnh v (gọi là nhãn của v).

– Gọi S là tập đỉnh đã xét.

– Khởi tạo: S = {0}, L(s0) =0, L(v)=vô cùng với mọi v thuộc V\S

– Tại mỗi bước lặp:

• Cập nhập lại nhãn các đỉnh thuộc V\S (tập V trừ tập S)

• Tìm đỉnh thuộc tập V\S có nhãn nhỏ nhất (tham lam) kề với S để đưa vào S.

Cập nhật nhãn L(v)

• Khởi tạo: S = {s0}, L(s0) =0, L(v)=vô cùng với mọi v thuộc V\S

Với mọi v thuộc V\S: Với mọi s thuộc S:

L(v) = min (L(v),L(s) +m(s,v))

Trong đó m(s,v) là độ dài đường đi từ s với v

• Vì chỉ có L(s\*) với s\* là đỉnh vừa duyệt xong ở bước trước là có thay đổi về giá trị nên việc tính lại L(v) chỉ có ý nghĩa với các đỉnh kề với s\*

Với mọi v thuộc V\S kề với s\*:

L(v) = min(L(v),L(s\*)+m(s\*,v))

Tìm đỉnh có nhãn nhỏ nhất s\*

• Đỉnh có nhãn nhỏ nhất s\*:

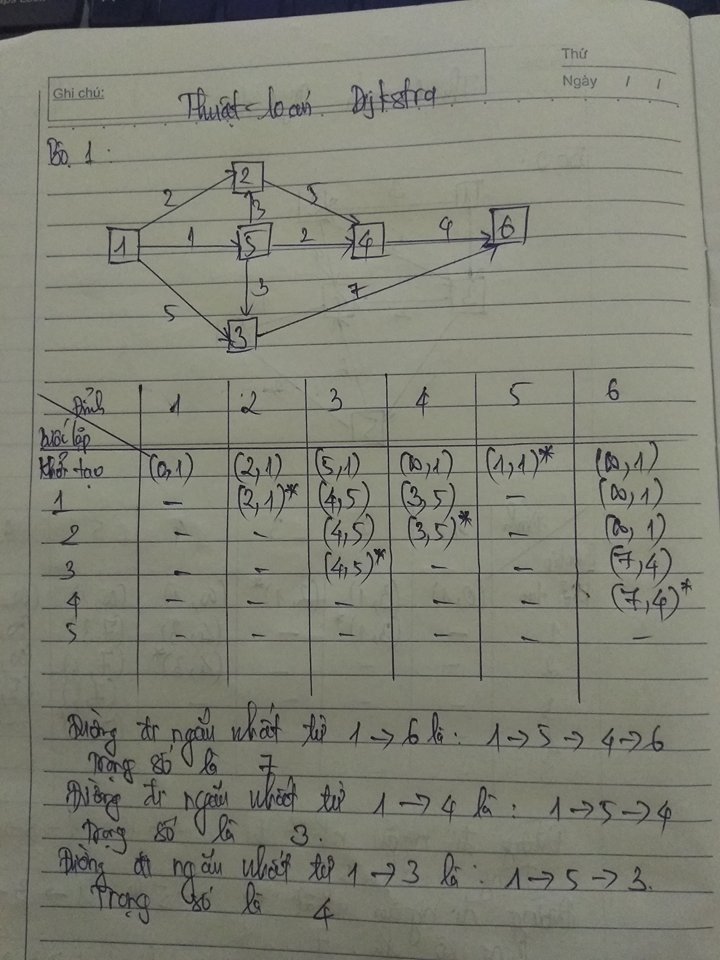
–Kề với 1 trong các đỉnh thuộc S Và

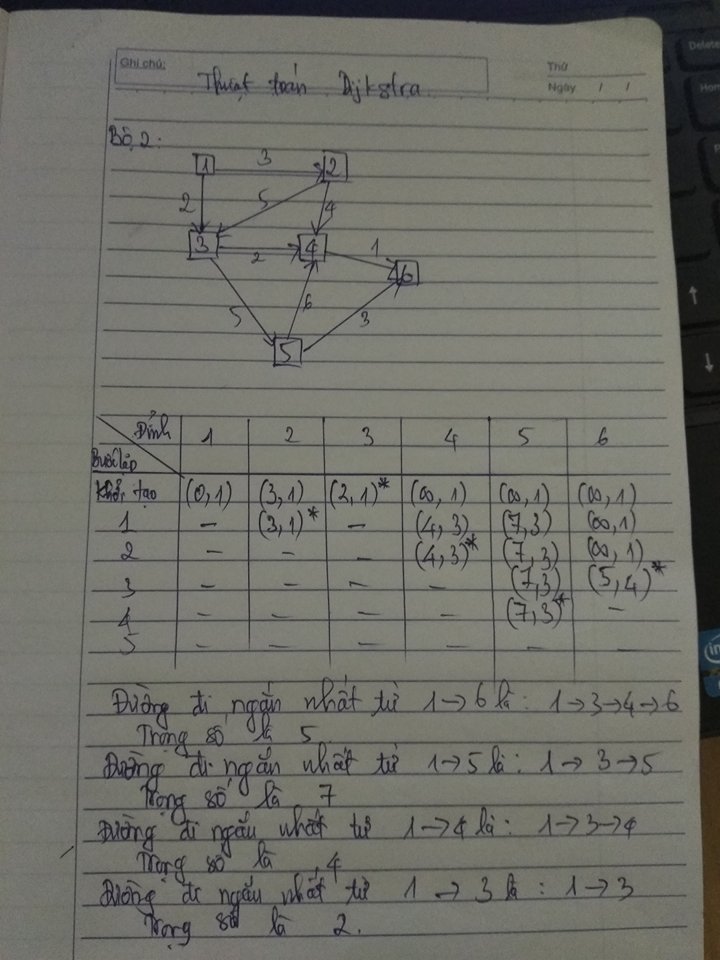
–L(s\*) = min(L(v): mọi v thuộc V\S)

1. **Độ phức tạp thuật toán.**

Thuật toán Dijkstra bình thường sẽ có độ phức tạp là O(n^2+m), do ta phải duyệt n lần (đối với n đỉnh), mỗi lần duyệt lại phải duyệt qua n đỉnh để tìm đỉnh có khoảng cách nhỏ nhất. Tuy nhiên ta có thể sử dụng kết hợp với cấu trúc heap hoặc set, khi đó độ phức tạp sẽ là O((m+n)log(n)), nếu dùng Fibonacci thì độ phức tạp giảm xuống còn O(m+nlogn). Trong đó m là số cạnh, n là số đỉnh của đồ thị đang xét

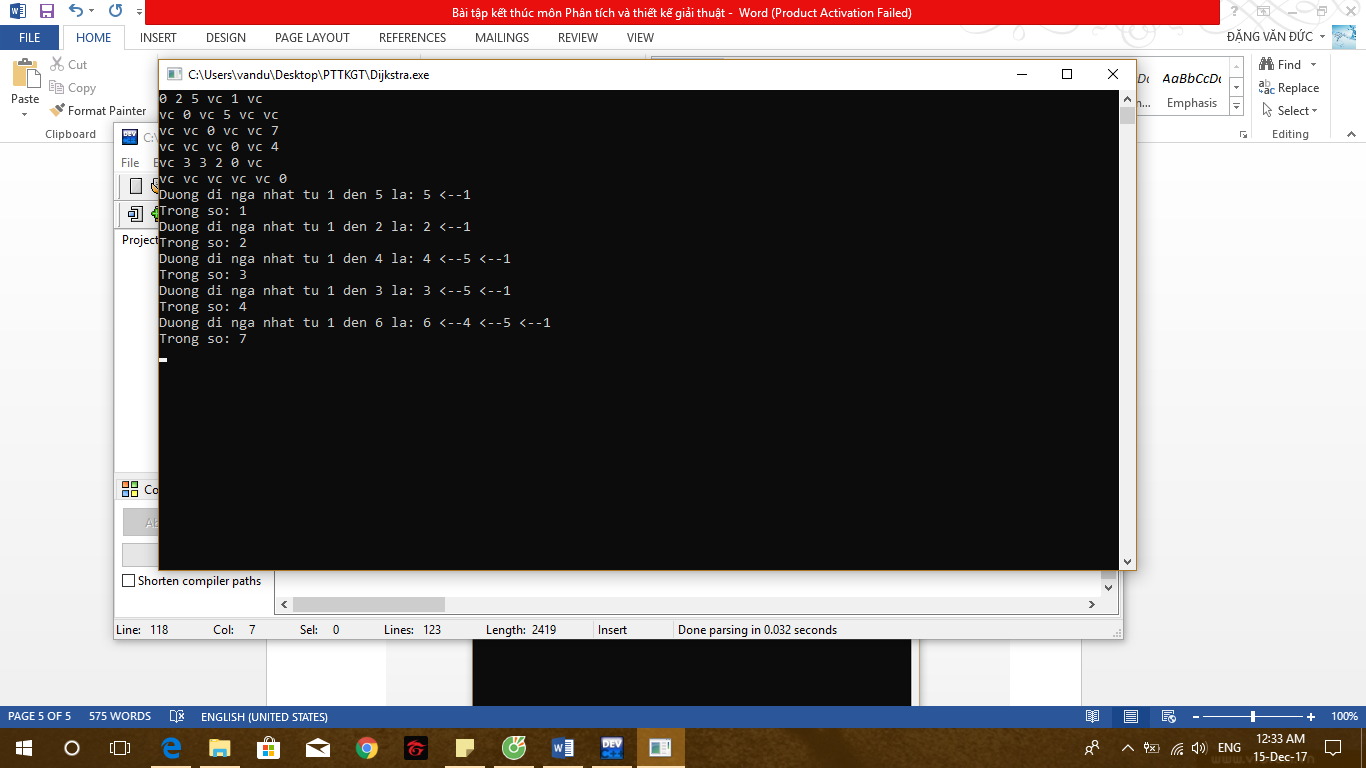
1. **Th ực hiện thuật toán với 2 bộ dữ liệu**

****

****

1. **Chương tình**
2. **Kết quả**

Bộ 1:



Bộ 2:

