

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



BÀI TẬP ĐỒ THỊ

Phan Nhật Tân - 23521405
Nguyễn Huy Phước - 23521234



Mục lục

1	Bài toán 1	3
2	Bài toán 2	4



1 Bài toán 1

Hãy tính toán chi phí đường đi nếu sử dụng thuật toán tìm kiếm Greedy và UCS. Viết chi tiết thông tin tuyến đường và đánh giá liệu các đường đi có tối ưu không.

Thuật toán Greedy Best-First Search

Quy tắc: Mở rộng node có giá trị heuristic nhỏ nhất.

Quá trình:

- Bắt đầu từ **London**: chọn **Hamburg**.
- Từ **Hamburg**: chọn **Falsterbo**.
- Từ **Falsterbo**: chọn **Danzig**.
- Từ **Danzig**: chọn **Visby**.
- Từ **Visby**: chọn **Talinn**.
- Từ **Talinn**: chọn **Novgorod**.

Đường đi: London → Hamburg → Falsterbo → Danzig → Visby → Talinn → Novgorod.

Tổng chi phí thực tế:

$$801 + 324 + 498 + 606 + 590 + 474 = 3293$$

Đánh giá: Đường đi không tối ưu vì chỉ dựa trên giá trị heuristic mà bỏ qua chi phí thực tế.

Thuật toán Uniform Cost Search (UCS)

Quy tắc: Mở rộng node có tổng chi phí thực tế nhỏ nhất.

Quá trình:

- Bắt đầu từ **London**: chọn **Amsterdam**.
- Từ **Amsterdam**: chọn **Hamburg**.
- Từ **Hamburg**: chọn **Lubeck**.
- Từ **Lubeck**: chọn **Danzig**.
- Từ **Danzig**: chọn **Visby**.
- Từ **Visby**: chọn **Riga**.
- Từ **Riga**: chọn **Talinn**.
- Từ **Talinn**: chọn **Novgorod**.

Đường đi: London → Amsterdam → Hamburg → Lubeck → Danzig → Visby → Riga → Talinn → Novgorod.

Tổng chi phí thực tế:

$$395 + 411 + 64 + 262 + 738 + 201 + 305 + 474 = 2850$$

Đánh giá: Đường đi là tối ưu vì thuật toán luôn chọn chi phí thực tế nhỏ nhất.



So sánh Greedy và UCS

- **Greedy:** Đường đi không tối ưu vì chỉ dựa trên giá trị heuristic.
- **UCS:** Đường đi tối ưu nhờ xét chi phí thực tế từ gốc.

2 Bài toán 2

Hãy xác định xem có chu trình âm trong đồ thị hay không để Kaiser có thể tiếp tục giải cứu vợ.

Ý tưởng và phương pháp thiết kế thuật toán

Ý tưởng chính

Sử dụng thuật toán Bellman-Ford để kiểm tra sự tồn tại của chu trình âm:

- Thực hiện $N - 1$ lần cập nhật khoảng cách ngắn nhất từ đỉnh xuất phát.
- Sau $N - 1$ lần, nếu còn cạnh nào làm giảm chi phí, thì tồn tại chu trình âm.
- Truy xuất chu trình âm bằng cách lần ngược qua mảng cha (*parent*).

Các bước thực hiện

1. Khởi tạo:

- Đặt giá trị ban đầu cho các đỉnh: $\text{dist}[u] = \infty$, ngoại trừ $\text{dist}[\text{start}] = 0$.
- Đặt mảng cha $\text{parent}[u] = -1$.

2. Cập nhật:

- Lặp $N - 1$ lần: Duyệt qua tất cả các cạnh (u, v, c) :
- Nếu $\text{dist}[u] + c < \text{dist}[v]$, thì cập nhật:

$$\text{dist}[v] = \text{dist}[u] + c, \quad \text{parent}[v] = u$$

3. Kiểm tra chu trình âm:

- Lặp thêm 1 lần: Nếu tồn tại cạnh (u, v, c) với $\text{dist}[u] + c < \text{dist}[v]$, thì chu trình âm tồn tại.
- Lần ngược qua mảng *parent* để truy xuất chu trình.

Mã giả

Input: Số đỉnh N , số cạnh M , danh sách cạnh $\text{edges}[a, b, c]$

Output: In ra YES và chu trình âm, hoặc NO nếu không có

1. Khởi tạo:

- $\text{dist}[u] = \text{INF}$ với mọi u , $\text{dist}[\text{start}] = 0$
- $\text{parent}[u] = -1$ với mọi u



2. Thực hiện cập nhật:

- Lặp $N-1$ lần:
 - Với mỗi cạnh (u, v, c) :
 - Nếu $\text{dist}[u] + c < \text{dist}[v]$:
 - $\text{dist}[v] = \text{dist}[u] + c$
 - $\text{parent}[v] = u$

3. Kiểm tra chu trình âm:

- Với mỗi cạnh (u, v, c) :
 - Nếu $\text{dist}[u] + c < \text{dist}[v]$:
 - Có chu trình âm
 - Truy xuất chu trình:
 - Tìm một đỉnh thuộc chu trình âm, lặp qua parent để tìm chu trình
 - In YES và chu trình, rồi kết thúc
- Nếu không tìm thấy cạnh nào giảm: In NO

Tính toán độ phức tạp

- Số lần duyệt cạnh: $O(N \times M)$.
- Truy xuất chu trình: $O(N)$.

Tổng độ phức tạp: $O(N \times M)$.