

Bài tập Lý thuyết 1

Thông tin sinh viên

Bài làm

Câu 1. Tìm kiếm đường đi từ S → G

1.1 Theo BFS

1.2 Theo DFS

Câu 2. Tìm kiếm đường đi từ A → G

2.1 Theo BFS

2.2 Theo DFS

Câu 3. Tìm đường đi từ S đến G trong mê cung dưới

3.1 Tìm kiếm theo chiều rộng

3.2 Tìm kiếm theo chiều sâu có kiểm tra trạng thái đang nằm trên đường đi để tránh lặp.

Thứ tự mở là Phải -> Dưới -> Trái -> Trên.

Thông tin sinh viên

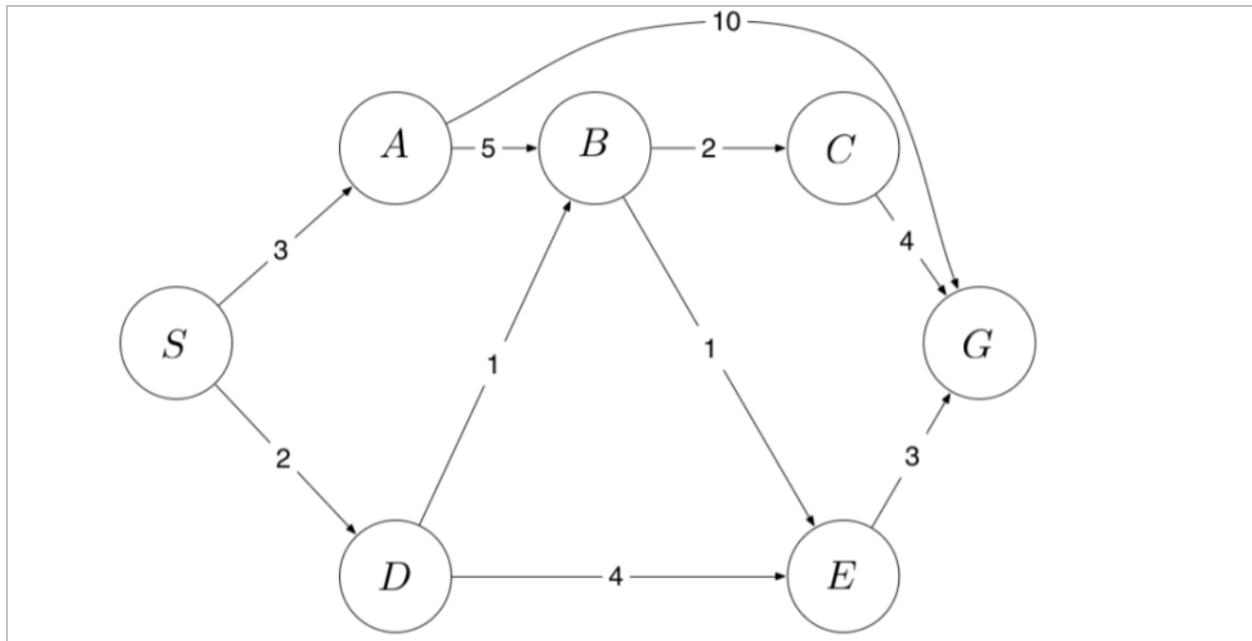
- MSSV: 22850034
- Họ và tên: Cao Hoài Việt
- Email: viet.ch2612@gmail.com

Bài làm

Câu 1. Tìm kiếm đường đi từ S → G



Ưu tiên đưa đỉnh vào Fringe theo thứ tự trên bảng chữ cái



1.1 Theo BFS

Fringe = [S] //Queue //Lấy S

Fringe = [A, D] //Lấy A

Fringe = [D, B, G] //Tìm thấy đích → Dừng tìm kiếm

Đường đi từ S tới G theo BFS sẽ là **S → A → G**

1.2 Theo DFS

Nếu sử dụng Stack

Fringe = [S] //Stack //Lấy S

Fringe = [A, D] //Lấy D

Fringe = [A, B, E] //Lấy E

Fringe = [A, B, G] //Lấy G → Tìm thấy đích, dừng tìm kiếm.

Đường đi từ S tới G theo DFS sẽ là **S → D → E → G**

Nếu sử dụng phương pháp chọn đỉnh theo thứ tự chữ cái. (Đệ quy)

S → A, D

Chọn A → B, G //Chọn B vì sẽ ưu tiên theo bảng chữ cái. DFS chưa tới G nên thuật toán chưa dừng lại.

Chọn B → C, E

Chọn C → G

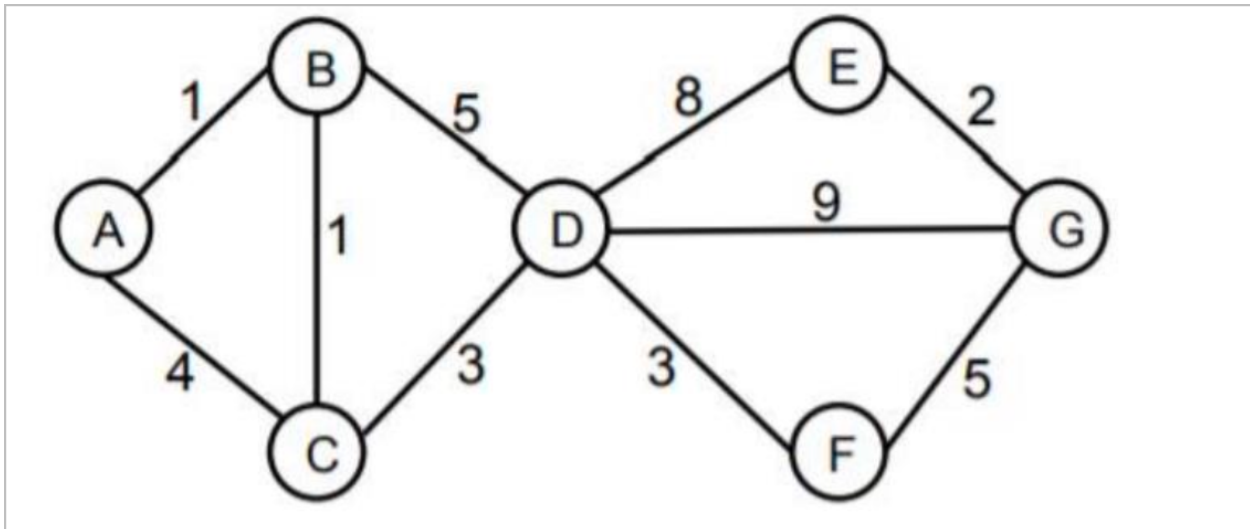
//Tìm thấy đích G → Dừng tìm kiếm

Đường đi từ S tới G theo DFS sẽ là S → A → B → C → G

Câu 2. Tìm kiếm đường đi từ A → G



Ưu tiên chọn đỉnh theo thứ tự trên bảng chữ cái



2.1 Theo BFS

Fringe = [A] //Queue //Lấy A

Fringe = [B, C] //Lấy B

Fringe = [C, D] //Lấy C

Fringe = [D] //Lấy D

Fringe = [E, F, G] //Tìm thấy đích G → Dừng tìm kiếm

Đường đi từ A tới G theo thuật toán BFS sẽ là A → B → D → G

2.2 Theo DFS

Nếu sử dụng Stack

Fringe = [A] //Stack //Lấy A

Fringe = [B, C] //Lấy C

Fringe = [B, D] //Lấy D

Fringe = [B, E, F, G] //Lấy G → Tìm thấy đích, dừng tìm kiếm.

Đường đi từ S tới G theo BFS sẽ là **A → C → D → G**

Nếu sử dụng phương pháp chọn đỉnh theo thứ tự chữ cái. (Đệ quy)

Chọn A → B, C

Chọn B → C, D

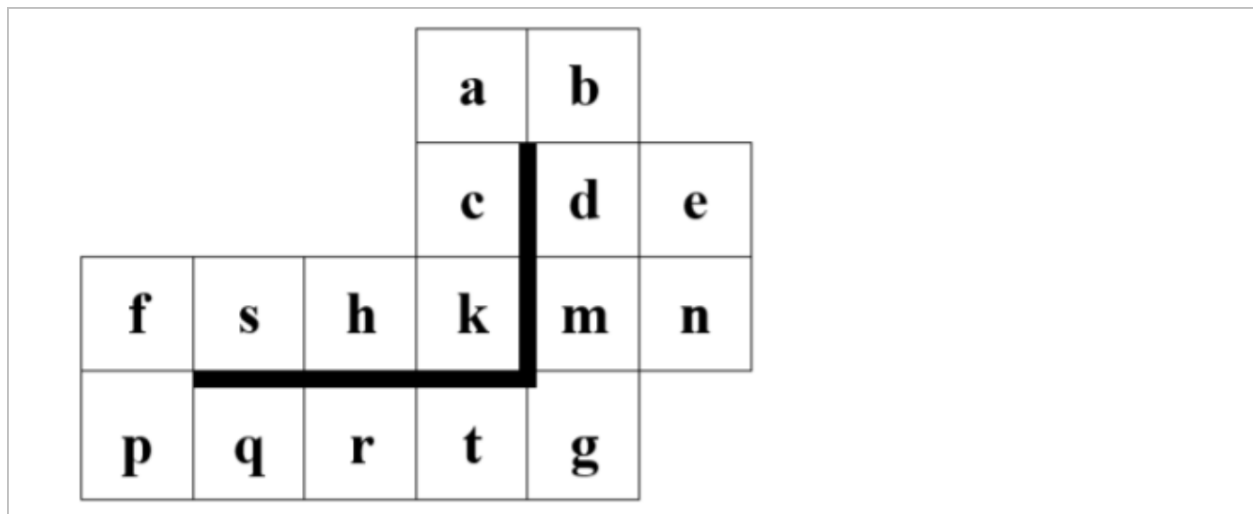
Chọn C → D

Chọn D → E, F, G

Chọn E → G //Tìm thấy đích → Dừng tìm kiếm

Đường đi từ A tới G theo thuật toán DFS sẽ là **A → B → C → D → E → G**

Câu 3. Tìm đường đi từ S đến G trong mê cung dưới



3.1 Tìm kiếm theo chiều rộng

Fringe = [S] //Queue //Lấy S

Fringe = [H, F] //Lấy H

Fringe = [F, K] //Lấy F

Fringe = [K, P] //Lấy K

Fringe = [P, C] //Lấy P

Fringe = [C, Q] //Lấy C

Fringe = [Q, A] //Lấy Q

Fringe = [A, R] //Lấy A

Fringe = [R, B] //Lấy R

Fringe = [B, T] //Lấy B

Fringe = [T, D] //Lấy T

Fringe = [D, G] //Tìm thấy G → Dừng thuật toán

Đường đi từ S đến G theo BFS sẽ là: **S → F → P → Q → R → T → G**

3.2 Tìm kiếm theo chiều sâu có kiểm tra trạng thái đang nằm trên đường đi để tránh lặp. Thứ tự mở là Phải -> Dưới -> Trái -> Trên.

Chọn **S** → **H**, **F**

Chọn **H** → **K**

Chọn **K** → **C**

Chọn **C** → **A**

Chọn **A** → **B**

Chọn **B** → **D**

Chọn **D** → **E**, **M**

Chọn **E** → **N** //Không có D vì D đã nằm trên đường đi

Chọn **N** → **M** //Không có E vì E đã nằm trên đường đi

Chọn **M** → **G** //Không có D, N vì đã nằm trên đường đi.

//Tìm thấy G → Dừng thuật toán

Đường đi từ S đến G theo DFS sẽ là: **S → H → K → C → A → B → D → E → N → M → G**