**BÁO CÁO THỰC HÀNH TUẦN 1**

Họ và Tên: Bùi Thị Thanh Phương

MSSV: 20110280

I. Nội Dung Thực Hành

Tìm đường đi ngắn nhất từ trường Đại học Khoa học Tự nhiên (V1) tới sân bay Tân Sơn Nhất (V18) dùng các thuật toán sau:

1. BFS

Ý tưởng:

* sử dụng một danh sách open(hang đợi) để lưu các trạng thái được sinh ra và chờ được phát triển
* Trạng thái nào được sinh ra trước sẽ được phát tiển trước, do đó danh sách open được xử lí như hàng đợi
* Đỉnh: đỉnh đang xét
* Open : các đình kề với đỉnh đang xét
* Close: Các đỉnh đã đi qua( được duyệt)

\*Lưu ý: Với đường đi có mũi tên và trọng số thì ta sẽ lưu ý hướng đi của nó, nếu có trọng số ta sẽ ưu tiên chọn trọng số nhỏ hơn, nếu không có trọng số ta sẽ ưu tiên chữ cái Amphabe

- Sau khi thực hiện xong quá trình ta sẽ dừng lại ngay vị trị cần tìm và tìm đường bằng cách đi ngược từ dưới bảng lên trên.

|  |  |  |  |
| --- | --- | --- | --- |
| Lần Lặp | Đỉnh | Open | Close |
| 0 | 0 | V1 |  |
| 1 | V1 | V2, V4, V3 | V1 |
| 2 | V2 | V4, V3, V7 | V1, V2 |
| 3 | V4 | V3, V7, V5 | V1, V2, V4 |
| 4 | V3 | V7, V5, V6, V8 | V1, V2, V4, V3 |
| 5 | V7 | V5, V6, V8 | V1, V2, V4, V3, V7 |
| 6 | V5 | V6, V8, V14 | V1, V2, V4, V3, V7, V5 |
| 7 | V6 | V8, V14, V12 | V1, V2, V4, V3, V7, V5, V6 |
| 8 | V8 | V14, V12, V10, V9 | V1, V2, V4, V3, V7, V5, V6, V8 |
| 9 | V14 | V12, V10, V9, V11 | V1, V2, V4, V3, V7, V5, V6, V8, V14 |
| 10 | V12 | V10, V9, V13 | V1, V2, V4, V3, V7, V5, V6, V8, V14, V12 |
| 11 | V10 | V9, V13 | V1, V2, V4, V3, V7, V5, V6, V8, V14, V12, V10 |
| 12 | V9 | V13, V11 | V1, V2, V4, V3, V7, V5, V6, V8, V14, V12, V10, V9 |
| 13 | V13 | V11 | V1, V2, V4, V3, V7, V5, V6, V8, V14, V12, V10, V9, V13 |
| 14 | V11 | V15 | V1, V2, V4, V3, V7, V5, V6, V8, V14, V12, V10, V9, V13, V11 |
| 15 | V15 | V16 | V1, V2, V4, V3, V7, V5, V6, V8, V14, V12, V10, V9, V13, V11, V15 |
| 16 | V16 | V17 | V1, V2, V4, V3, V7, V5, V6, V8, V14, V12, V10, V9, V13, V11, V15, V16 |
| 17 | V17 | V18 | V1, V2, V4, V3, V7, V5, V6, V8, V14, V12, V10, V9, V13, V11, V15,V16, V17 |
| 18 | V18 |  | V1, V2, V4, V3, V7, V5, V6, V8, V14, V12, V10, V9, V13, V11, V15, V15, V17, V18 |

Vậy: Đường đi ngắn nhất: V1-V3-V8-V9-V11-V15-V16-V17-V18 tương ứng với vị trí thứ ( 0, 2, 7, 8, 10, 14, 15, 16, 17)

1. DFS

Ý tưởng:

* Tại các trạng thái được chọn để phát triển là trạng thái sinh ra sau cùng trong số trạng thái chờ phát triển
* Danh sách open các trạng thái chờ tương tự ngăn xếp (stack)
* Đỉnh nào vào sau thì sẽ lấy ra để xét trước. Khi them vào Open thì them vào bên trái => lấy đỉnh xét tiếp thì lấy đỉnh ngoài cùng bên trái.

\*Lưu ý: Với đường đi có mũi tên và trọng số thì ta sẽ lưu ý hướng đi của nó, nếu có trọng số ta sẽ ưu tiên chọn trọng số nhỏ hơn, nếu không có trọng số ta sẽ ưu tiên chữ cái Amphabe

Sau khi thực hiện xong quá trình ta sẽ dừng lại ngay vị trị cần tìm và tìm đường bằng cách đi ngược từ dưới bảng lên trên.

|  |  |  |  |
| --- | --- | --- | --- |
| Lần lặp | Đỉnh | Open | Close |
| 0 | 0 | V1 |  |
| 1 | V1 | V2, V4, V3 | V1 |
|  | V2 | V7, V4, V3 | V1, V2 |
|  | V7 | V4, V3 | V1, V2, V7, |
|  | V4 | V5, V3 | V1, V2, V7, V4 |
|  | V5 | V14, V3 | V1, V2, V7, V4, V5 |
|  | V14 | V11, V3 | V1, V2, V7, V4, V5, V14 |
|  | V11 | V15, V10, V3 | V1, V2, V7, V4, V5, V14, V11 |
|  | V15 | V16, V10, V3 | V1, V2, V7, V4, V5, V14, V11 |
|  | V16 | V17, V10, V3 | V1, V2, V7, V4, V5, V14, V11, V16 |
|  | V17 | V18, V10, V3 | V1, V2, V7, V4, V5, V14, V11, V16, V17 |
|  | V18 |  |  |

Vậy đường đi: V1-V4-V5-V14-V11-V15-V16-V17-V17 tương ứng với vị trí thứ ( 0, 3, 4, 13, 10, 14, 15, 16, 17)

1. **UCS**

Ý tưởng: Thuật toán UCS là một thuật toán duyệt, tìm kiếm trên một cấu trúc cây, hoặc đồ thị có trọng số (chi phí). Việc tìm kiếm bắt đầu tại nút gốc và tiếp tục bằng cách duyệt các nút tiếp theo với trọng số hay chi phí thấp nhất tính từ nút gốc. UCS sử dụng một hàng đợi ưu tiên (Priority Queue – PQ) để lưu trữ và duyệt các trạng thái trên đường đi.

PQ= {(DHKHTN(V1), 0)}

PQ= { V2(50), V4(300), V3(350)}

PQ= {V4(300), V3(350), V7(650)}

PQ= {V3(350), V7(650), V5(1600)}

PQ= {V6(450), V7(650), V8(1250), V5(1600)}

PQ= {V7(650), V12(1150), V8(1250), V5(1600)}

PQ= {V12(1150), V8(1250), V5(1600) }

PQ= {V8(1250), V5(1600), V13(2100) }

PQ= {V5(1600), V9(2040), V13(2100)}

PQ= {V9(2040), V13(2100), V14(3000)}

PQ={V13(2100), V14(3000), V11(3240)}

PQ={V14(2700), V11(3240)}

PQ={V11(3240)}

PQ={V15(3640), V10(4040)}

PQ={V10(4040), V16(4940)}

PQ={V17(5710)}

PQ= {V18(6910)}

Vậy đường đi: V1-V3-V8-V11-V15-V16-V17-V18 tương ứng với vị trí thứ (0, 2, 7, 10, 14, 15, 16, 17)

II. Thực hành

Các lỗi đã sữa:

* Tại hàm cài đặt thuật toán BFS



Lỗi: khai báo hàm là In thường

Sữa lại:



* Tại hàm DFS



Lỗi: khai báo hàm là In thường

Sữa lại

A picture containing text

Description automatically generated

* Sai vị trí hàm Main

Sữa: Đổi vị trí hàm main xuống dưới

* Lỗi khai báo **“adjList”**

Text

Description automatically generated

Sữa lại

Text

Description automatically generated

* **Đã khai báo thêm phần Graph**