

Họ tên: Đỗ Thành Nhon
MSSV: 1512387
Lớp: CNTN2015

CƠ SỞ TRÍ TUỆ NHÂN TẠO

BÁO CÁO ĐỒ ÁN 1

BÀI TẬP 1

Giảng viên lý thuyết: Lê Hoài Bắc
Giảng viên hướng dẫn thực hành:
Lê Ngọc Thành
Nguyễn Ngọc Thảo
Nguyễn Hải Minh

Các mốc thực hiện đồ án:

Giai đoạn	Công việc
1	Tìm hiểu các thuật toán BFS, DFS, UCS, GBFS và A*.
2	Thiết kế cấu trúc dữ liệu lưu trữ và thuật toán sử dụng.
3	Tiến hành cài đặt.
4	Viết các Test case, tiến hành kiểm tra và sửa lỗi (nếu có).
5	Viết báo cáo.

I. Thiết kế lưu trữ đồ thị và các thông tin liên quan.

1. Thiết kế đồ thị.

- Đồ thị N đỉnh sẽ được lưu trữ trong mảng 2 chiều có kích thước $N * N$. Trong đó $Matrix[i][j] = 0$ nếu không có đường đi từ i tới j và $Matrix[i][j] = a$ nếu tồn tại đường đi từ i tới j với chi phí là a .
- Thông tin về giá trị Heuristic sẽ được lưu trong mảng một chiều có kích thước N phần tử trong đó mỗi phần tử $Array[i]$ là giá trị Heuristic của đỉnh thứ i .

2. Thiết kế chương trình.

Chương trình được viết theo phương pháp lập trình hướng đối tượng.

```
class Graph{
private:
    int n;
    int **a;
    int *heuristic;
    int start;
    int end;
    int* trace;
    vector<int> visit;
public:
    Graph();
    Graph(const Graph&);
    Graph(int, int**, int*);
    ~Graph();
    void set(istream &f);
    void bfs();
    void dfs();
    void ucs();
    void gbfs();
    void astar();
    void print(ostream&);
};
```

Giải thích lựa chọn các thuộc tính:

N : lưu số lượng đỉnh của đồ thị.

- Mảng hai chiều a : lưu các giá trị thể hiện cạnh và trọng số của đồ thị.
- Mảng một chiều heuristic: Lưu các giá trị heuristic.
- Số nguyên start: đỉnh bắt đầu.
- Số nguyên end: đỉnh kết thúc.
- Mảng một chiều trace: $trace[v] = u$ đánh dấu từ đỉnh u có đường đi đến đỉnh v .
- Mảng một chiều visit: Lưu các đỉnh đã mở.

Giải thích lựa chọn các phương thức:

- `Graph()`: constructor.
- `Graph(const Graph&)`: constructor.
- `Graph(int, int**, int*)`: constructor.
- `~Graph()`: destructor.

- Phương thức set: dùng để đọc input và xây dựng đồ thị.
- Phương thức bfs: Hàm tìm kiếm theo chiều rộng.
- Phương thức dfs: Hàm tìm kiếm theo chiều sâu.
- Phương thức ucs: Hàm tìm kiếm chi phí đồng nhất.
- Phương thức gbfs: Hàm tìm kiếm tham lam.
- Phương thức astar: Hàm tìm kiếm A*.

II. Ví dụ chương trình.

1. Ví dụ 1.

Input

```
6
0 5
0 2 3 0 5 0
2 0 0 4 0 0
3 0 0 0 4 0
0 4 0 0 1 2
5 0 4 1 0 5
0 0 0 2 5 0
6 2 5 2 1 0
```

Output

BFS

```
0 1 2 4 3 5
0 4 5
```

DFS

```
0 1 3 4 2 5
0 1 3 4 5
```

UCS

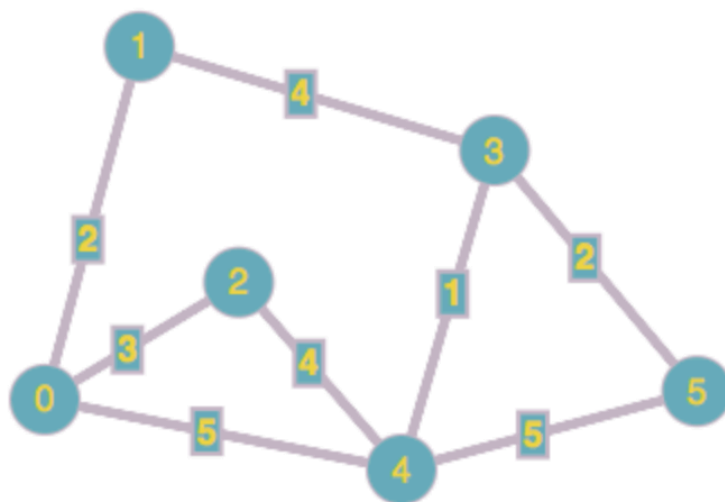
```
0 1 2 4 3 5
0 1 3 5
```

GBFS

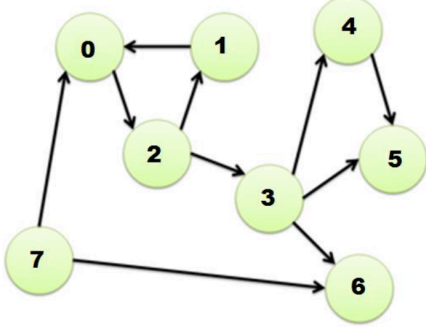
```
0 4 5
0 4 5
```

Astar

```
0 1 4 2 3 5
0 1 3 5
```



2. Ví dụ 2.

Đồ thị	input.txt
	8 1 6 0 0 1 0 0 0 0 0 1 0 0 0 0 0 0 0 0 1 0 1 0 0 0 0 0 0 0 0 1 1 1 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 1 0 5 6 4 2 8 7 0 2

Output

BFS

```
6 4 5 7 2 3 1 0
6 4 2 0
```

DFS

```
6 4 2 0
6 4 2 0
```

UCS

```
6 4 3 7 2 5 1 0
6 4 7 1 0
```

GBFS

```
6 5 1 0
6 5 1 0
```

Astar

```
6 4 3 7 1 5 0
6 4 7 1 0
```

III. Các lưu ý khi chạy chương trình.

- Phiên bản Visual studio: 2017.
- Ngôn ngữ sử dụng: C++.
- Chương trình được gọi bằng tham số dòng lệnh như sau:
1512387_1_1.exe <ten_file_input>
- Cấu trúc file Input như trong yêu cầu đề bài.
- Output gồm 5 file ghi lại kết quả theo từng thuật toán.