


Trường: ĐH CNTP TP.HCM Khoa: Công nghệ thông tin Bộ môn: Khoa học máy tính MH: TH Cấu trúc rời rạc	Chương 2. Cây	
---	----------------------	---

A. MỤC TIÊU:

- Cài đặt thuật toán nhập đồ thị từ file
- Cài đặt thuật toán duyệt đồ thị theo chiều sâu và chiều rộng
- Ứng dụng thuật toán duyệt đồ thị vào bài toán đếm số thành phần liên thông

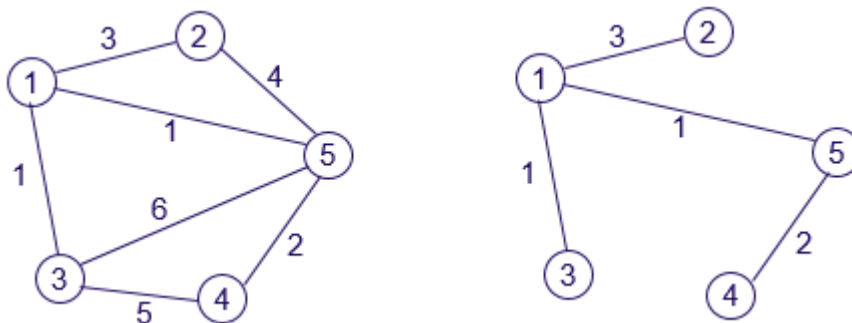
B. DỤNG CỤ - THIẾT BỊ THÍ NGHIỆM CHO MỘT SV:

STT	Chủng loại – Quy cách vật tư	Số lượng	Đơn vị	Ghi chú
1	Computer	1	1	

C. NỘI DUNG THỰC HÀNH

I. Tóm tắt lý thuyết

- Cây là một đồ thị vô hướng liên thông không có chu trình.
- Cho $G = (V, E)$ là đồ thị vô hướng liên thông. Cây khung nhỏ nhất của G là cây $T = (V, F)$ với $F \subset E$ và có tổng trọng số các cạnh của T là nhỏ nhất.
- Ví dụ minh họa:



- Có nhiều phương pháp tìm kiếm cây khung nhỏ nhất như: Kruskal và Prim.

A. Thuật toán Kruskal

Ý tưởng:

Bước 1: Sắp các cạnh theo thứ tự tăng dần của trọng số, $T = \emptyset$

Bước 2: Lần lượt duyệt trong danh sách cạnh đã sắp xếp theo thứ tự trọng số từ nhỏ đến lớn, chọn cạnh bổ sung vào tập T với điều kiện việc bổ sung này không tạo thành chu trình.

Bước 3: Tiếp tục thực hiện bước 2 cho đến khi T có n-1 cạnh thì dừng.

B. Thuật toán Prim

Ý tưởng:

Bước 1: Gọi T là tập chứa các cạnh của cây khung; $T=\emptyset$. S là tập chứa các đỉnh của cây khung; $S=\{x\}$ (với x là đỉnh bất kỳ thuộc đồ thị)

Bước 2: Tìm cạnh $(x,y) \in w(s)$ sao cho cạnh này có trọng số bé nhất trong $w(s)$.

$$T=T\cup(x,y); S=S\cup\{y\}$$

Bước 3: Lặp bước 2 đến khi T có n-1 cạnh thì dừng.

II. Bài tập hướng dẫn mẫu

Bài 1. Cho đồ thị G có trọng số, liên thông gồm n đỉnh được đánh số từ 1 đến n. Hãy áp dụng thuật toán Prim tìm cây khung nhỏ nhất tương ứng cho đồ thị G.

```
#include "stdio.h"
#include "conio.h"
#include "stdlib.h"
#define vc 100
#define vmax 20

//Định nghĩa cấu trúc đồ thị
struct dothi
{
    int flag;           //loại đồ thị: 0 là vô hướng, 1 là có hướng
    int n;              //số đỉnh của đồ thị
    int w[vmax][vmax]; //ma trận kề hoặc ma trận trọng số
};

int chuaxet[vmax]; //định nghĩa cấu trúc đồ thị
struct dothi
{
    int flag;           //loại đồ thị: 0 là vô hướng, 1 là có hướng
    int n;              //số đỉnh của đồ thị
    int w[100][100];    //ma trận kề hoặc ma trận trọng số
};

//Đọc dữ liệu từ file lưu vào cấu trúc đồ thị g
```

```

void docfile(dothi &g)
{
    FILE *f = fopen("d:\\dt_vohuong_1.txt", "r");
    if (f == NULL)
        printf("\nKhong tim thay file");
    else
    {
        fscanf(f, "%d", &g.flag);
        fscanf(f, "%d", &g.n);
        for (int i = 1; i <= g.n; i++)
            for (int j = 1; j <= g.n; j++)
                fscanf(f, "%d", &g.w[i][j]);
    }
}

//Xuất thông tin đồ thị
void xuatdothi(dothi g)
{
    if (g.flag == 0)
        printf("\nDo thi vo huong");
    else
        printf("\nDo thi co huong");
    printf("\nMa tran ke hoac ma tran trong so cua do thi:");
    for (int i = 1; i <= g.n; i++)
    {
        printf("\n");
        for (int j = 1; j <= g.n; j++)
            printf("%5d", g.w[i][j]);
    }
}

```

```

//Phần tạo cây khung nhỏ nhất theo thuật
//Định nghĩa cấu trúc cạnh
struct nhan
{
    int dinh;
    int chiphi;
};

//xuất bảng thực hiện thuật toán Prim
void xuatbangchiphi(dothi g, nhan bangchiphi[vmax][vmax])
{
    printf("\nBang mo ta thuat toan Prim tim cay khung nho nhat \n");
    for (int i = 1; i <= g.n; i++)
        printf("%10d", i);

    for (int i = 0; i <= g.n; i++)
    {
        printf("\n-----\n");
        for (int j = 1; j <= g.n; j++)
            printf("\t(%d, %d)", bangchiphi[i][j].dinh,
                bangchiphi[i][j].chiphi);
        getch();
    }
}

//xuất cây khung nhỏ nhất
void xuatcaykhung(dothi g, nhan bangchiphi[vmax][vmax])
{
    printf("\nDanh sach cac canh thuoc cay khung nho nhat.\n");
    int i = 2;
    while (i <= g.n)
    {
        printf("\t(%d, %d)", bangchiphi[g.n][i].dinh, i);
        i++;
    }
}

//xây dựng cây khung theo thuật toán Prim
void Prim(dothi g, nhan bangchiphi[vmax][vmax], int dinhkhoidau)
{
    int chuaxet[vmax];

```

```

//bước khởi tạo
for(int i = 1; i <= g.n; i++)
{
    chuaxet[i] = 0;
    bangchiphi[0][i].chiphi = vc;
    bangchiphi[0][i].dinh = 0;
}
bangchiphi[0][dinhkhoidau].chiphi = 0;
bangchiphi[0][dinhkhoidau].dinh = 0;

int i = 0;
while(i < g.n)
{
    int chondinh_min = 1;
    while (chuaxet[chondinh_min] == 1)
        chondinh_min++;

    for (int j = 1; j < g.n; j++)
    {
        if (chuaxet[j] == 0 && bangchiphi[i][j].chiphi <
            bangchiphi[i][chondinh_min].chiphi)
            chondinh_min = j;
    }
    //printf("\n===== %d", chondinh_min);
    //getch();
    chuaxet[chondinh_min] = 1;
    i++;
    for (int j = 1; j <= g.n; j++)
    {
        if (chuaxet[j] == 1 || g.w[chondinh_min][j] == vc)
            bangchiphi[i][j] = bangchiphi[i - 1][j];
        if (chondinh_min != j && chuaxet[j] == 0 &&
            g.w[chondinh_min][j] < vc)
            if (bangchiphi[i - 1][j].chiphi > g.w[chondinh_min][j])
            {
                bangchiphi[i][j].chiphi = g.w[chondinh_min][j];
                bangchiphi[i][j].dinh = chondinh_min;
            }
        else
            bangchiphi[i][j] = bangchiphi[i - 1][j];
    }
}

```

```

    }
}

void main()
{
    dothi g;
    docfile(g);
    xuatdothi(g);
    nhan bangchiphithi[vmx][vmx];
    Prim(g, bangchiphithi, 1);
    xuatbangchiphithi(g,bangchiphithi);
    xuatcaykhung(g, bangchiphithi);
    getch();
}

```

Bảng kết quả:

D:\demo_lythuyetdothi\Debug\demo_lythuyetdothi.exe

```

Do thi vo huong
Ma tran ke hoac ma tran trong so cua do thi:
  0  2  2  2  4 100 100 100
  2  0 100 100 100  1 100 100
  2 100  0 100  3 100 100  1
  2 100 100  0  4  3 100 100
  4 100  3  4  0 100  7 100
100  1 100  3 100  0  5 100
100 100 100 100  7  5  0  6
100 100  1 100 100 100  6  0
Bang mo ta thuat toan Prim tim cay khung nho nhat
      1          2          3          4          5          6          7          8
-----
(0,0)  (0,100) (0,100) (0,100) (0,100) (0,100) (0,100) (0,100)
-----
(0,0)  (1,2)  (1,2)  (1,2)  (1,4)  (0,100) (0,100) (0,100)
-----
(0,0)  (1,2)  (1,2)  (1,2)  (1,4)  (2,1)  (0,100) (0,100)
-----
(0,0)  (1,2)  (1,2)  (1,2)  (1,4)  (2,1)  (6,5)  (0,100)
-----
(0,0)  (1,2)  (1,2)  (1,2)  (3,3)  (2,1)  (6,5)  (3,1)
-----
(0,0)  (1,2)  (1,2)  (1,2)  (3,3)  (2,1)  (6,5)  (3,1)
-----
(0,0)  (1,2)  (1,2)  (1,2)  (3,3)  (2,1)  (6,5)  (3,1)
-----
(0,0)  (1,2)  (1,2)  (1,2)  (3,3)  (2,1)  (6,5)  (3,1)
-----
(0,0)  (1,2)  (1,2)  (1,2)  (3,3)  (2,1)  (6,5)  (3,1)
Danh sach cac canh thuoc cay khung nho nhat.
(1, 2) (1, 3) (1, 4) (3, 5) (2, 6) (6, 7) (3, 8)

```

III. Bài tập ở lớp

Bài tập 2. Cài đặt thuật toán Kruskal tìm cây khung nhỏ nhất.

IV. Bài tập về nhà

Bài tập 3. Giả sử có một cơn bão tuyết đi ngang qua một thành phố nọ (thành phố này có n quận khác nhau) làm cho hệ thống đường bộ ở đây phủ đầy tuyết. Một đội cứu hộ muốn thực hiện cào tuyết để giúp người dân có thể lưu thông từ quận này sang quận khác. Hãy giúp đội cứu hộ cào như thế nào sao cho nhanh nhất?

Gợi ý: Thiết kế một bản đồ mô tả các cung đường lưu thông giữa các quận trong thành phố. Mỗi cung đường có 1 chiều dài nhất định. Vận dụng một trong hai thuật toán trên để giải quyết bài toán.

Bài tập 4. Bài toán tìm mạng điện với độ tin cậy lớn nhất. Cho lưới điện có n nút. Đường dây nối nút i với nút j có độ tin cậy $1 > p[i, j] > 0$ với $i, j = 1, 2, \dots, n$. Gọi $G = (V, E)$ là đồ thị tương ứng với lưới điện này. Hãy tìm cây khung H của đồ thị G với độ tin cậy lớn nhất.