

Trường: ĐH CNTP TP.HCM Khoa: Công nghệ thông tin Bộ môn: Khoa học máy tính MH: TH Cấu trúc rời rạc	<h1>Chương 2.</h1> <h2>Cây</h2>	
---	---------------------------------	--

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

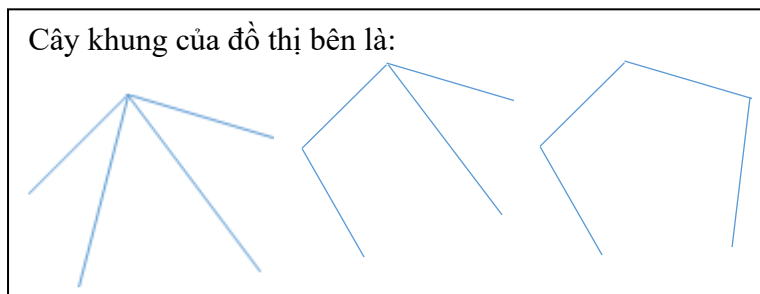
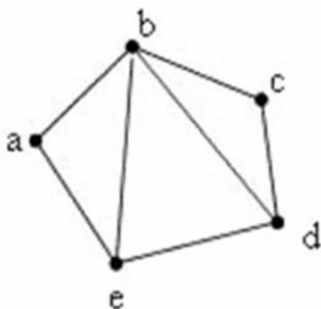
B. DỤNG CỤ - THIẾT BỊ THÍ NGHIỆM CHO MỘT SV:

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 của G là cây $T = (V, F)$ với $F \subset E$.
- Ví dụ minh họa: cho đồ thị như hình vẽ



- Có nhiều phương pháp tìm kiếm cây khung như: sử dụng thuật toán DFS hoặc BFS.

A. Thuật toán DFS (Depth First Search)

Ý tưởng:

- Xuất phát từ 1 đỉnh v chưa thăm cho trước.
- Thăm đỉnh v và tìm đỉnh u chưa thăm và kề với v . Nếu tồn tại thì thăm u , ngược lại quay lui về đỉnh gần với v nhất mà tại đó tìm được đỉnh kề chưa thăm. Lặp lại bước này cho đến khi không tìm được đỉnh kề hoặc tất cả các đỉnh đã được thăm.

B. Thuật toán BFS (Breadth First Search)

Ý tưởng:

- Xuất phát từ 1 đỉnh v chưa thăm cho trước, cho v vào hàng đợi
- Lấy đỉnh u từ hàng đợi, thăm u và lưu các đỉnh kề chưa thăm của u vào hàng đợi

- Lặp lại bước trên cho đến khi hàng đợi rỗng.

II. Bài tập hướng dẫn mẫu

Cho đồ thị G liên thông có n đỉnh, được đánh số từ 1 đến n. Áp dụng thuật toán DFS xây dựng cây khung tương ứng cho đồ thị G.

```
#include "stdio.h"
#include "conio.h"
#include "stdlib.h"
#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
};

//đọc dữ liệu từ file lưu vào cấu trúc đồ thị
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
//Định nghĩa cấu trúc cạnh
struct canh
{
    int u, v;
};

int chuaxet[100]; //đánh dấu các đỉnh đã xét hay chưa, đầu tiên khởi tạo các đỉnh chưa xét

//xuất các cạnh của cây khung
void xuatcaykhung(canh T[], int socanh)
```

```

{
    printf("\nCay khung nhỏ nhất \n");
    for (int i = 1; i <= socanh; i++)
        printf("\t(%d, %d)", T[i].u, T[i].v);
}

//xây dựng cây khung theo thuật toán DFS
void xaydungcaykhung_DFS(dothi g, canh T[], int &socanh, int u)
{
    chuaxet[u] = 1;
    for (int v = 1; v <= g.n; v++)
    {
        if (g.w[u][v] != 0 && chuaxet[v] == 0)
        {
            socanh++;
            T[socanh].u = u;
            T[socanh].v = v;

            if (socanh == g.n - 1)
            {
                xuatcaykhung(T, socanh);
                return;
            }
            else
                xaydungcaykhung_DFS(g, T, socanh, v);
        }
    }
}

//xây dựng cây khung theo thuật toán BFS
void xaydungcaykhung_BFS(dothi g, canh T[], int &socanh)
{
    for (int u = 1; u <= g.n; u++)
    {
        for (int v = 1; v <= g.n; v++)
        {
            if (g.w[u][v] != 0 && chuaxet[v] == 0)
            {
                socanh++;
                T[socanh].u = u;
                T[socanh].v = v;
                chuaxet[v] = 1;
                chuaxet[u] = 1;
                printf("\n===== (%d,%d)", T[socanh].u, T[socanh].v);
                if (socanh == g.n - 1)
                {
                    xuatcaykhung(T, socanh);
                    return;
                }
            }
        }
    }
}

void main()
{
    dothi g;
    docfile(g);
    xuatdothi(g);

    canh T[100];
    int socanh = 0;
    for (int i = 1; i <= g.n; i++)

```

```

        chuaxet[i] = 0;
        xaydungcaykhung_DFS(g, T, socanh, 1);
        //socanh = 0;
        //xaydungcaykhung_BFS(g, T, socanh);
        getch();
    }

```

```

Do thi vo huong
Ma tran ke hoac ma tran trong so cua do thi:
    0    1    1    2    0
    1    0    1    0    0
    1    1    0    1    1
    2    0    1    0    0
    0    0    1    0    0
Cay khung
      (1,2)  (2,3)  (3,4)  (3,5)

```

III. Bài tập ở lớp

Bài 1. Hãy cài đặt các thuật toán duyệt khác như: DFS không đệ quy, BFS để tìm các cây khung của đồ thị.

Bài 2. Giả sử có một cơn bão tuyết đi ngang qua một thành phố n (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. Em hãy giúp đội cứu hộ vạch ra các cung đường cần cào.

IV. Bài tập về nhà

Bài 3. Ứng dụng cây khung cho bài toán tìm đường đi trong bản đồ.