**BÁO CÁO BÀI TẬP THỰC HÀNH 4**

**Cây khung nhỏ nhất với Krukal**

* Cách tổ chức thiết kế lớp đối tượng

|  |  |
| --- | --- |
| **CGraph** |  |
| Nhan : Vector<int>  Label : int  SoPTLienThong : Vector<int>  NhanDinh : Vector<int>  vector<CEDGE> lstEdge;  int tongTrongSo; | Mảng các đỉnh gán nhãn ( cùng nhãn sẽ LT )  Số nhãn gán  Số phần tử liên thông ( đỉnh )  Số đỉnh liên thông với nhau ( đếm lại )  Danh sách cạnh được input  Tổng trọng số của 1 cây khung nhỏ nhất |
| arrayInit : Vector<int>  KTDoThiDon : Bool  SapDoThi :  Visit :  ThanhPhanLienThong:  XuatThanhPhanLT:  IsAdjacentMatValid:  PrimAlg()  PrimAlgEx()  XuatCayKhung()  InitListEdge(CMatrix );  SortListEdge();  KruskalAlg(CGraph C, int n);  HoanViCanh(int, int);  XuatCayKhungKruskal  UpdateLabels  SortListEdgeEx()  TimTPLT  KiemTraDoThiLT | Thay đổi size của Mảng với giá trị n mới  Kiểm tra đồ thị đơn hay ko, trả về true hoặc false  Sắp đồ thị để khi xuất ra sẽ theo thứ tự yêu cầu của Thầy  Hàm Visit dùng để kiểm tra và gán các đỉnh liên thông  Hàm để gán nhãn các đỉnh liên thông với nhau  Xuất ra file ra in ra màn hinh console các TP liên thông  Kiểm tra tính hợp lệ ma trận kề  Hàm tìm cây khung nhỏ nhất với thuật toán Prim  Hàm tìm cây khung phần mở rộng theo yêu cầu  Hàm xuất cây khung ra file và Console  Tao ra danh sách cạnh từ 1 ma trận  Sắp xếp danh sách cạnh theo thứ tự yêu cầu  Ham tim cay khung nho nhat bang thuat toan Kruskal  Hoán vị 2 cạnh  Xuất cây khung Kruskal  Gán nhãn 2 đỉnh  Sắp cạnh tăng dần theo đỉnh  Hàm tìm thành phần LT trả về 1 vector<Graph>  Kiểm tra đồ thị LT |

|  |  |
| --- | --- |
| **CMatrix** |  |
|  |  |
| charArray2D array2D  matrixInit  <<, >>  GetSize() | Ma trận  Thay đổi size của ma trận với giá trị n mới  Viết operator Cin Cout cho ma trận  Lấy Size của ma trận |

|  |  |
| --- | --- |
| **CEDGE** |  |
| int v;//Dinh thu nhat  int w;//Dinh thu hai  int \_trongSo; | Đỉnh đầu của 1 cạnh  Định cuối của cạnh  Trọng số của cạnh |
| int LayV();int LayW();  int LayTrongSo();  void GanV(int );  void GanW(int );  void GanTrongSo(int ) | Hàm lấy đỉnh đầu, cuối và trọng số của cạnh  Hàm gán giá trị cho đỉnh đầu, cuối và Trọng số cạnh |

|  |  |
| --- | --- |
| **CSpanningTree** | **Lớp cây khung ( chứa nhiều cạnh Edge)** |
|  |  |
| vector <CEDGE> Tree;  void AddEdge(CEDGE ); | 1 SpanningTree là 1 mảng động gồm nhiều cạnh  Hàm Add 1 cạnh vào 1 SpanningTree |

* Ý tưởng thuật toán Krukal tìm cây khung nhỏ nhất :

Tạo 1 mảng các cạnh có trọng số tăng dần từ ma trận cho trước

Gán nhãn các đỉnh theo thứ tự ( vd đỉnh 0 nhãn 0, đỉnh 1 nhãn 1)

V là tập các cạnh, V=0;

Tim các cạnh e thỏa V+{e} thì ko tạo thành chu trình

Dùng kỹ thuật gán nhãn để xét xem có chu trình hay ko

Điều kiện dừng khi xét đến soCanh -1 hoặc V có soDinh-1 cạnh

* Ý tưởng về phần nâng cao

Xử lý trong phần xuất cây khung Krukal

Xét xem nếu đồ thì có liên thông hay ko?

* Nếu có thì xuất theo yêu cầu
* Nếu ko liên thông ta sử dụng hàm tìm các TPLT trả ra 1 vector<Cgraph>

Chạy thuật toán xuất cây khung Krukal với vector<Cgraph> vừa tìm được.

* Ý tưởng về hảm Liên thông trả ra 1 vector<Cgraph>

Sử dụng hàm tìm liên thông củ ta sẽ tim được 2 TPLT, mỗi TPLT sẽ ứng với danh sách các định

Tạo vector <int> để lưu các đỉnh này lại

Tạo tiếp 1 biến < Cgraph > temp để lưu tổ hợp các cạnh từ các đỉnh này ( chỉ lấy các cạnh có trọng số) và sau đó sử dụng vector<Cgraph> để push\_back (temp)