**BÁO CÁO BÀI TẬP THỰC HÀNH 3**

**Cây khung nhỏ nhất với Prim**

* Cách tổ chức thiết kế lớp đối tượng

|  |  |
| --- | --- |
| **CGraph** |  |
| Nhan : Vector<int>  Label : int  SoPTLienThong : Vector<int>  NhanDinh : Vector<int> | 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 ) |
| arrayInit : Vector<int>  KTDoThiDon : Bool  SapDoThi :  Visit :  ThanhPhanLienThong:  XuatThanhPhanLT:  IsAdjacentMatValid:  PrimAlg()  PrimAlgEx()  XuatCayKhung() | 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 |

|  |  |
| --- | --- |
| **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 ( là 1 mảng 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 Prim cây khung nhỏ nhất :

Ta đánh dấu đỉnh bất kỳ là 1 của 1 mảng được tạo và đánh dấu là 0 ( để làm cờ ). Ta xét khi mà số cạnh của cây khung nhỏ hơn số cạnh đồ thì trừ 1, duyệt trên ma trận được input vào ta xét những đỉnh nối với đỉnh bắt đầu để tìm trọng số nhỏ nhất trong số các cạnh đó, sau đó ta thêm cạnh vào cây khung. Đánh dấu đỉnh cuối của cạnh lấy sẽ là định bắt đầu kế típ, tìm cho đến hết.

* Ý tưởng về phần nâng cao
* Tạo 1 ma trận mới giống y như ma trận input vào
* Biến ma trân đó thành ma trận kề ko trọng số ( mục đích là tìm được số thanh phần liên thông và cách đỉnh liên thông với nhau )
* Tìm 1 đỉnh trong 1 thành phần liên thông , sử dụng hàm Prim mở rộng ( ta đánh dấu đỉnh bắt đầu là đỉnh trong Thành phần liên thông )
* Phần còn lại vẫn sử dụng thuật tìm kiếm như thuật toàn Prim thường