| **NAME:** | Kunal Bhatia |
| --- | --- |
| **UID:** | 2021300010 |
| **SUBJECT** | DAA |
| **EXPERIMENT NO:** | 9 |
| **AIM:** | To implement Approximation algorithms |
| **Algorithm:** | **Vertex Cover Problem:**  1) Initialize the result as {}  2) Consider a set of all edges in given graph. Let the set be E.  3) Do following while E is not empty   1. Pick an arbitrary edge (u, v) from set E and add 'u' and 'v' to result. 2. Remove all edges from E which are either incident on u or v.   4) Return result |
| **Code:** | #include<iostream>  #include<set>  #include<vector>  #include<time.h>  using namespace std;  struct Edge {      char u;      char v;      Edge() {          u = 'x';          v = 'x';      }      Edge(char u, char v) {          this->u = u;          this->v = v;      }  };  set<char> vertexCover(vector<Edge> &edges) {      set<char> cover;      vector<Edge> copy\_edges(edges);      srand(time(NULL));      while(!copy\_edges.empty()) {          int i = rand() % copy\_edges.size();          Edge e = copy\_edges[i];          copy\_edges.erase(copy\_edges.begin() + i);          cover.insert(e.u);          cover.insert(e.v);          cout<<"Adding edge "<<e.u<<" "<<e.v<<endl;          for (int i = 0; i < copy\_edges.size(); i++) {              if(copy\_edges[i].u == e.u || copy\_edges[i].v == e.u || copy\_edges[i].u == e.v || copy\_edges[i].v == e.v) {                  cout<<"Removing edge "<<copy\_edges[i].u<<" "<<copy\_edges[i].v<<endl;                  copy\_edges.erase(copy\_edges.begin() + i);                  i--;              }          }      }      return cover;  }  int main() {      cout<<"Enter the number of edges: ";      int n;      cin>>n;      cout<<"Enter edges (u v):"<<endl;      vector<Edge> edges(n);      for(int i = 0; i < n; i++) {          char u, v;          cin>>u>>v;          edges[i] = Edge(u,v);      }      set<char> cover = vertexCover(edges);      cout<<"\nVertex cover: {";      for(char v : cover) {          cout<<v<<", ";      }      cout<<"}"<<endl;      return 0;  } |
| **Output:** |  |
| **Conclusion:** | After performing this experiment, I have understood approximation algorithms and implemented it to solve the vertex cover problem. |