### 并查集

#include<stdio.h>  
#include<stdlib.h>  
#define VERTICES 6  
  
void initialize(int parent[], int rank[]) {  
 int i;  
 for (i = 0; i < VERTICES; i++) {  
 parent[i] = -1;  
 rank[i] = 0;  
 }  
}  
  
int find\_root(int x, int parent[]) {  
 int x\_root = x;  
 while (parent[x\_root] != -1) {  
 x\_root = parent[x\_root];  
 }  
 return x\_root;  
}  
  
/\* 1 - union successfully, 0 - union failed \*/  
int union\_vertices(int x, int y, int parent[], int rank[]) {  
 int x\_root = find\_root(x, parent);  
 int y\_root = find\_root(y, parent);  
  
 if (x\_root == y\_root) {  
 return 0;  
 }  
 else {  
 //parent[x\_root] = y\_root;  
 if (rank[x\_root] > rank[y\_root]) {  
 parent[y\_root] = x\_root;  
 }  
 else if (rank[y\_root] < rank[x\_root]) {  
 parent[x\_root] = y\_root;  
 }  
 else {  
 parent[x\_root] = y\_root;  
 rank[y\_root]++;  
 }  
 return 1;  
 }  
}  
  
int main()  
{  
 int parent[VERTICES] = {0};  
 int rank[VERTICES] = {0};  
 int edges[6][2] = {  
 {0, 1}, {1, 2}, {1, 3},  
 {2, 4}, {3, 4}, {2, 5}  
 };  
  
 initialize(parent, rank);  
 int i;  
 for (i = 0; i < 6; i++) {  
 int x = edges[i][0];  
 int y = edges[i][1];  
 if (union\_vertices(x, y, parent, rank) == 0) {  
 printf("Cycle detected!\n");  
 exit(0);  
 }  
 }  
 printf("No cycles fond.\n");  
  
 return 0;  
}