## 一般模式：

.h文件的命名空间中写函数声明：

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 并查集 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

namespace HuangYuan {

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 查找树根 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int find(int X);

};

.cpp文件的命名空间中写变量声明定义和函数体：

namespace HuangYuan {

const int MAX\_N = 10000;

int par[MAX\_N]; // 节点的父节点

int rank[MAX\_N]; // 数的高度

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* 查找树根 \*/

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

int find(int X) {

return X == HuangYuan::par[X] ? X : find(HuangYuan::par[X]);

}

}

## 算法技巧：

### 1、一般深度优先模式：

for(int i=1; i<=n; i++) {

if(vt[i] == 0) {

vt[i] = 1;

…….

dfs(t+1);

vt[i] = 0;

}

}

这种代码模式相当于排列。深度优先遍历不一定全部用这个模式实现。

当代码运行很久都不出结果时，尝试利用判断来剪枝。

### 2、物品选不选的情况全部罗列出来

cur += score[t];

vt[t] = 1;

dfs(t+1, cur);

cur -= score[t];

vt[t] = 0;

dfs(t+1, cur);

这种代码模式相当于物品选不选的情况全部罗列出来

### 3、全排列：

代码模式：

for(int i=t; i<10; i++) {

tem=arr[t]; arr[t]=arr[i]; arr[i]=tem;

dfs(t+1);

tem=arr[t]; arr[t]=arr[i]; arr[i]=tem;

}

4、m = a\*a + b\*b + c\*c + d\*d，则暴力破解的时候，可以让a<sqrt(m)为止。

（a < sqrt(m)）

### 5、快速排序中选择轴的函数：

int posi(int p, int r) {

int i = p;

int j = r+1;

int tem = p;

while(i<j) {

while((i<j)&&(arr[++i] < arr[tem])) ;

while(arr[--j] > arr[tem]);

if(i>=j) break;

swrap(i, j);

}

swrap(tem, j);

return j;

}

6、 当出现几个变量的值同时不知道的情况下，可以把每个变量都变成一个循环，利用循环嵌套暴力破解出答案。

7、想让深度优先遍历的结果不重复，可以让结果有序。例如，想从12个数中选出5个数，可以从小到大逐一选出，选出的结果自然就是有序的。又如1，2，3，4，5和5，3，1，2，4选出的其实是一样的。有序选出使得不重复。

8、交换次数最少的排序算法：选择排序

9、如果要让除的结果包含小数，可以在分子那里乘上一个1.0

如：(a\*1.0)/b

### 10、蚂蚁碰头问题：

（1）如2014-08的题目，蚂蚁碰头，然后互相往反方向走，和蚂蚁穿行是一样的。

### 11、跳台阶问题：

（1）小明刚刚看完电影《第39级台阶》，离开电影院的时候，他数了数礼堂前的台阶数，恰好是39级!

站在台阶前，他突然又想着一个问题：

如果我每一步只能迈上1个或2个台阶。先迈左脚，然后左右交替，最后一步是迈右脚，也就是说一共要走偶数步。那么，上完39级台阶，

有多少种不同的上法呢？

（2）加入现在是走到39级台阶，前面一次只能是到37（39-2）级 或者 38（39-1）级，依次类推。从而引入递归思想，f(n) = f(n-1) + f(n-2); 模拟行走方案，当级数走完了，判断步数是否满足即可。

### 12、除法问题：

1、在判断中遇到除法，要变成乘法判断，因为整数除整数，会变成整数，比如5/2 的结果为2，本身5/2应该不等于2的，但是，计算机会把5/2和2作为相等的结果。从而引起错误。

### 13、平面寻找连通点问题：

（1）解决方法：上下左右四个方向，寻找新的（没有被访问过的）的点，查看该点是不是在集合里面（之前是被选中的点），然后把该点标记为已访问，找到个数+1，如果最后找到个数为集合的点的个数，则该集合是连通分量。

（2）总结：一次深度优先遍历能够找到所有在集合里面的点，则该集合是连通分量。