二分搜索：

bool binary\_search(int value)

{

int l = 0, r = 6;

while(l <= r)

{

int m = (l+r) / 2;

if(num[m] < value)

l = m + 1;

else if(num[m] > value)

r = m - 1; // +1 -1避免死循环

else

return true;

}

return false;

}

输出所有字典序：

int main()

{

cin>>n;

while(n--)

{

cin>>s;

sort(s.begin(), s.end());

cout<<s<<endl;

while(next\_permutation(s.begin(), s.end()))

cout<<s<<endl;

cout<<endl;

}

}

memset()函数原型是extern void \*memset(void \*buffer, int c, int count)

头文件cstring

buffer：为指针或是数组,

c：是赋给buffer的值,

count：是buffer的长度.

这个函数在socket中多用于清空数组.如:原型是memset(buffer, 0, sizeof(buffer))

Memset 用来对一段内存空间全部设置为某个字符，一般用在对定义的字符串进行初始化

而fill函数可以赋值任何，而且使用方法特别简便：

fill(arr, arr + n, 要填入的内容);

头文件algorithm

fill 初始化二维数组 fill(arr[0], arr[0]+n\*n, 0);

并查集：

int pre[1000];

int find(int x) //查找根结点

{

int r = x;

while(pre[r] != r)

r = pre[r];

int i = x, j; //路径压缩

while(i != r)

{

j = pre[i]; //在改变上级之前用临时变量 j 记录下他的值

pre[i] = r; //把上级改为根节点

i = j;

}

return r;

}

void join(int x, int y)

{

int fx = find(x);

int fy = find(y);

if(fx != fy)

pre[fx] = fy;

}

for(int i=0; i<n; i++) // 初始化

pre[i] = i;