解题报告

## Q.C：语法百题 排列组合

题目：**组合序列，组合一般是指从n（<=10)个不同元素中取r（<=n）个元素的不同组合的数目。按字典序输出。**

解题思路：用深度优先搜索在状态树上从第一层开始，到了第n层（约束条件）终止，如果没到第n层就继续递归调用该函数。本题在这个基础上还要让答案不能有重复的数，而且后面的数要比前面的数打，所以另定义一个bool函数（use）来判断这个数是否用过，而且限制每个答案只输出r个数，这就是解空间。

主函数中dns（1）中“1”的意义：表示是从状态树中的第一层开始搜索。

需要注意的三点：解空间，约束条件，状态树。

正确代码：

//#include <bits/stdc++.h>

#include <iostream>

using namespace std;

int n,m;

int a[100];

bool use[100];

void dfs(int deep){

if(deep>m){

for(int i=1; i<=m; i++) cout<<a[i]<<" ";

cout<<endl;

return;

}

for(int i=a[deep-1]+1; i<=n; i++){

a[deep]=i;

if(!use[i]){

use[i]=true;

dfs(deep+1);

use[i]=false;

}

}

}

int main()

{

ios::sync\_with\_stdio(false);

cin>>n>>m;

dfs(1);

return 0;

}

解题报告

## Q.A：语法百题 二分查找

题目：输入N，M。  
然后输入N个整数a[1]...a[N]。  
然后输入M个整数b[1]..b[M]，判断b[i]是否出现在a[i]中，若在输出"yes"，否则输出"no"。  
(1<=N<=10^5，1<=M<=10^5，1<=a[i]<=10^9，1<=b[i]<=10^9)

解题思路： 先定义一个bool的分治的函数，里面变量为一个用来查找的数组a，要找的数key，从数组下标l到r。在里面先定义mid为数组中间的数的下标。然后判断a[mid]是否就是key，如果是就说明找到答案了，就可以返回true，如果没有就判断key是大于a[min]还是小于，然后从另一半里再找中间的数对比key。一直重复，直到最后如果一直没有找到答案，就返回false。然后在main中先输入除了要查找的几个key外所有的量，然后sort排序一下，再分别for循环输入每个key，输入一个之后就用分治函数，判断返回值为true，就输出yes，否则输出no，并且换行；

需要注意题意是要把所有的key分别在上面的数组里面查找是否存在。

正确代码：

#include <bits/stdc++.h>

using namespace std;

int a[1000000];int n,m;

bool b\_s(int \*a,int key,int l,int r){

int left=l,right=r,mid;

while(left<right){

mid=(left+right)>>1;

// cout<<"KEY= "<<key<<" l= "<<left<<" r= "<<right<<" mid= "<<mid<<" "<<endl;

// for(int i=1; i<=n; i++) cout<<"NO. "<<i<<" a["<<i<<"]= "<<a[i]<<endl;

if(a[mid]==key) return true;

if(key>a[mid]) left=mid+1; else right=mid;

}

return false;

}

bool cmp(int x,int y){return x<y;}

int main()

{

scanf("%d %d",&n,&m);

for(int i=1; i<=n; i++) scanf("%d",&a[i]);

sort(a+1,a+n+1,cmp);

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

int t; scanf("%d",&t);

if(!b\_s(a,t,1,n)) printf("no\n"); else printf("yes\n");

} return 0;}

解题报告

## Q.C：语法百题 排列组合

题目：**组合序列，组合一般是指从n（<=10)个不同元素中取r（<=n）个元素的不同组合的数目。按字典序输出。**

解题思路：用深度优先搜索在状态树上从第一层开始，到了第n层（约束条件）终止，如果没到第n层就继续递归调用该函数。本题在这个基础上还要让答案不能有重复的数，而且后面的数要比前面的数打，所以另定义一个bool函数（use）来判断这个数是否用过，而且限制每个答案只输出r个数，这就是解空间。

主函数中dns（1）中“1”的意义：表示是从状态树中的第一层开始搜索。

需要注意的三点：解空间，约束条件，状态树。

正确代码：

//#include <bits/stdc++.h>

#include <iostream>

using namespace std;

int n,m;

int a[100];

bool use[100];

void dfs(int deep){

if(deep>m){

for(int i=1; i<=m; i++) cout<<a[i]<<" ";

cout<<endl;

return;

}

for(int i=a[deep-1]+1; i<=n; i++){

a[deep]=i;

if(!use[i]){

use[i]=true;

dfs(deep+1);

use[i]=false;

}

}

}

int main()

{

ios::sync\_with\_stdio(false);

cin>>n>>m;

dfs(1);

return 0;

}

解题报告

## Q.C：

题目：对于给定的一个长度为 N的正整数数列 A[i]，现要将其分成 **M**（**M<=N**）段，并要求每段连续，且每段和的最大值最小。

关于最大值最小：

例如一数列 4 2 4 5 1要分成 3段

将其如下分段：

[4 2][4 5][1]

第一段和为 6，第 2段和为 9，第 3段和为 1，和最大值为 9。

将其如下分段：

[4][2 4][5 1]

第一段和为 4，第 2段和为 6，第 3段和为 6，和最大值为 6。

并且无论如何分段，最大值不会小于 6。

所以可以得到要将数列 4 2 4 5 1要分成 3段，每段和的最大值最小为 6。

解题思路： 把二分查找直接放在主函数中使用，前面定义一个bool函数judge（int sum）来判断将a数组进行数列分段，分段后最小值可不可以为sum，再通过主函数中的二分来节省时间。

正确代码：

#include <bits/stdc++.h>

using namespace std;

int n,m,a[1000000];

bool judge(int sum){

int tot=0,cur=0;

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

if(a[i]+cur>sum) {tot++; cur=a[i];}

else if(a[i]+cur==sum) {tot++; cur=0;}

else if(a[i]+cur<sum) cur+=a[i];}

if(cur) tot++;

if(tot<=m) return true;

return false;

}

int main(){

scanf("%d %d",&n,&m); int maxx=0;

for(int i=1; i<=n; i++){scanf("%d",&a[i]); maxx=max(maxx,a[i]);}

int l=maxx,r=2000000000,mid;

while(l<r){

mid=(l+r)>>1;

if(judge(mid)) r=mid;

else l=mid+1;

}

printf("%d\n",l);

return 0;}

解题报告

## Q.B：

题目：仙境的居民们决定举办一场程序设计区域赛。裁判委员会完全由自愿组成，他们承诺要组织一次史上最公正的比赛。他们决定将选手的电脑用星形拓扑结构连接在一起，即将它们全部连到一个单一的中心服务器。为了组织这个完全公正的比赛，裁判委员会主席提出要将所有选手的电脑等距离地围绕在服务器周围放置。

为购买网线，裁判委员会联系了当地的一个网络解决方案提供商，要求能够提供一定数量的等长网线。裁判委员会希望网线越长越好，这样选手们之间的距离可以尽可能远一些。

该公司的网线主管承接了这个任务。他知道库存中每条网线的长度（精确到厘米），并且只要告诉他所需的网线长度（精确到厘米），他都能够完成对网线的切割工作。但是，这次，所需的网线长度并不知道，这让网线主管不知所措。

你需要编写一个程序，帮助网线主管确定一个最长的网线长度，并且按此长度对库存中的网线进行切割，能够得到指定数量的网线。

解题思路： 先定义一个judge(int sum)用来判断是否可以得到m根sum长的网线，如果可以返回true，否则返回false，然后输入需要的数值，为了方便先把输入的数\*100，最后答案再/100。然后从0到a数组中最大的数进行二分，用judge来判断最大答案在哪，最后输出l-1在换算成米/100.

正确代码：#include <bits/stdc++.h>

using namespace std;

int n,m,a[1000000];

bool judge(int num){

if(num==0) return true;

int ans=0;

for(int i=1; i<=n ;i++)ans+=a[i]/num;

if(ans>=m) return true;

return false;

}

int main(){

scanf("%d %d",&n,&m); int maxx=-1;

for(int i=1; i<=n; i++){double t; scanf("%lf",&t); a[i]=t\*100; maxx=max(maxx,a[i]);}

sort(a+1,a+n+1);

int l=0,r=maxx,mid;

while(l<r){

mid=(l+r)>>1;

if(judge(mid)) l=mid+1;

else r=mid;

}

double s=l-1;

cout<<"l="<<l<<" l-1="<<l-1<<" s="<<s<<endl;

printf("%.2lf\n",s\*1.00/100.00); return 0;}

栈

头文件:#include <stack>

定义:stack <type> s;

用法:

s.push(参数)往栈里压一个

s.pop()将栈顶弹出一个

s.top()返回值是栈最上面的数

s.empty()返回bool类型的数，如果栈是空的返回true，否则返回false

s.size()栈长度

队列:

头文件:#include <queue>

定义:queue <type> q;

用法:

q.push(参数)向队列里面放一个参数

q.pop()弹出队列最前面的数

q.front()队列里最前面的数

q.back()队列里最后面的数

q.empty()返回bool类型的数，如果栈是空的返回true，否则返回false

q.size()队列长度