**最大乘积**

问题描述

　　对于n个数，从中取出m个数，如何取使得这m个数的乘积最大呢？

输入格式

　　第一行一个数表示数据组数  
　　每组输入数据共2行：  
　　第1行给出总共的数字的个数n和要取的数的个数m，1<=n<=m<=15，  
　　第2行依次给出这n个数，其中每个数字的范围满足:a[i]的绝对值小于等于4。

输出格式

　　每组数据输出1行，为最大的乘积。

样例输入

1  
5 5  
1 2 3 4 2

样例输出

48

C++代码：

#include "iostream"

#include "stdio.h"

#include "string"

#include "math.h"

#include "ctype.h"

#include "vector"

#include "stdlib.h"

#include "string.h"

#include "time.h"

#include "set"

#include "algorithm"

using namespace std;

int ans=-(1<<29);

int n,m;

int a[1000];

void dfs(int \*a,int cur,int sum,int k)

{

if(k>m)

return ;

if(cur==n)

{

if(k==m)

ans=std::max(ans,sum);

return ;

}

dfs(a,cur+1,sum\*a[cur],k+1);

dfs(a,cur+1,sum,k);

}

int main()

{

int T;

cin>>T;

while(T--)

{

cin>>n>>m;

for (int i = 0; i < n; i++)

{

cin>>a[i];

}

ans=-(1<<29);

dfs(a,0,1,0);

cout<<ans<<endl;

}

return 0;

}

C代码：

#include<stdio.h>

#include<stdlib.h>

#include<string.h>

int max;

void fun(int \*val,int index,int size,int m,int cnt,int res);

int main()

{

int n,m,x,i;

int val[15];

scanf("%d",&x);

while(x--)

{

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

for(i=0;i<n;i++)

{

scanf("%d",&val[i]);

}

max = -10000000;

fun(val,0,n,m,0,1);

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

}

return 0;

}

void fun(int \*val,int index,int size,int m,int cnt,int res)

{

if(m==cnt)

{

if(res > max)

{

max = res;

}

return ;

}

if(index>=size)

{

return ;

}

fun(val,index+1,size,m,cnt+1,res\*val[index]);

fun(val,index+1,size,m,cnt,res);

}

JAVA代码：

import java.util.Arrays;

import java.util.Scanner;

public class Main {

int s = 0x80000000;

public void input() {

Scanner in = new Scanner(System.in);

int t = in.nextInt();

while (t > 0) {

s = 0x80000000;

t--;

int n = in.nextInt();

int m = in.nextInt();

int[] num = new int[n];

for (int i = 0; i < n; i++) {

num[i] = in.nextInt();

}

count(num, 0, m, n,1);

int min = num[0];

System.out.println(s);

}

}

private void count(int[] num,int i,int m,int n,int sum){

if(m==0&&sum>s)s=sum;

if(i<n&&m>=0){

count(num,i+1,m-1,n,sum\*num[i]);

count(num,i+1,m,n,sum);

}

}

public static void main(String[] args) {

new Main().input();

}

}