**《算法设计与分析》实验报告**

# 实验一 递归与分治策略

|  |  |  |  |
| --- | --- | --- | --- |
| **报告书** | | | |
| 姓名 | 吴宇敖 | 指导教师 |  |
| 学号 | 176001752 | 日 期 |  |
| 班级 | 17计算机软件一班 |  |  |
| **实验内容** | | | |
| **[要求先在OJ上提交通过]**   1. 一组n个数，要从中选出其中m个数，使得选出的m个数之间的最小间隔最大。   提示，如果假设间隔为1，然后不断+1测试，直到最大可能，计算量太大；所以可以采用二分。   1. （请用递推方式求解）对于一个2行N列的走道。现在用1\*2,2\*2的砖去铺满。问有多少种不同的方式。下图是一个2行17列的走道的某种铺法。 IMG_256   思考：观察前n个结果，可以得到递推式子；如果N很大，需要高精度计算 | | | |
| **实验目的** | | | |
| 1）了解二分查找的方法  2）使用递归进行分析后，使用递推进行编程  3）使用精确计算的方法，可以到达高位数 | | | |
| **实验过程和步骤** | | | |
| **【实验题目】**  一组n个数，要从中选出其中m个数，使得选出的m个数之间的最小间隔最大。提示，如果假设间隔为1，然后不断+1测试，直到最大可能，计算量太大；所以可以采用二分。  **【程序代码及注释】**  #include<iostream>  #include<algorithm>  using namespace std;  bool cheak(int \*num,int mid,int n,int m)//进行查找，是否可以选出m个满足要求的数  {  int res=1;//满足要求的数  int k;  k=num[0];  for(int i=1;i<n;i++)  {  if(num[i]-k>=mid)//找到一个满足要求的res加1  {  res++;  k=num[i];  }  if(res>=m)  break;  }  if(res>=m)  return true;  else  return false;  }  int find(int \*num,int n,int m)//二分计算  {  int high=num[n-1]-num[0];  int low=0;  int mid;  while(low<=high)  {  mid=(high+low)/2;//计算中间值  if(cheak(num,mid,n,m))  low=mid+1;  else high=mid-1;  }  return low-1;  }  int main()  {  int n,m;  int interval;  int num[2000];  cin>>n>>m;  for(int i=0;i<n;i++)//初始化ˉ  {  cin>>num[i];  }  sort(num,num+n);//排序  interval=find(num,n,m);  cout<<interval<<endl;  //system("pause");  return 0;  }  **【实验结果及遇到的问题】（附输入输出）**    问题：  （1）在oj上进行提交时，出现runtime error错误，进询问老师后才发现需要注释system("pause");这个代码。  （2）进行二分查找是，一开始没有将low加1，导致无法正确的跳出循环，最后会是值变为0  **【算法分析】**  本实验使用了二分查找的方法，可以快速的查找到目标值。对于每一个值，都进行判定是否可以成功的找到数量为m个数，如果成功，将low改为这个值加1，在根据low和high进行下一个值的计算。  **【实验题目】**  （请用递推方式求解）对于一个2行N列的走道。现在用1\*2,2\*2的砖去铺满。问有多少种不同的方式。下图是一个2行17列的走道的某种铺法。 IMG_256  思考：观察前n个结果，可以得到递推式子；如果N很大，需要高精度计算  **【程序代码及注释】**  #include<iostream>  using namespace std;  int num[2000][1000];//最多2000列，100位数  int main()  {  int n;  for(int i=0;i<2000;i++)//初始化ˉ  for(int j=0;j<1000;j++)  num[i][j]=0;  num[1][1]=1;  num[2][1]=3;  int k=1;//表示有几位，方便后面进行输出  cin>>n;//输入有几列  for(int i=3;i<=n;i++){  for(int j=1;j<=k;j++)  num[i][j]=num[i-1][j]+num[i-2][j]\*2;//计算下一列的某一位  for(int j=1;j<=k;j++){  num[i][j+1]+=num[i][j]/10;//进位  if(num[i][j]>=10&&j==k)//增加一位  k++;  num[i][j]%=10;  }  }  for(int i=k;i>=1;i--)//输出  {  cout<<num[n][i];  }  cout<<endl;  //system("pause");  return 0;  }  **【实验结果及遇到的问题】（附输入输出）**      问题：  （1）一开始没有使用高精度计算，在列数大于一定程度是发现无法得出结果，之后使用了高精度计算才解决了问题。  （2）在进行输出时，一开始输出的就够是反的，导致一直找不到错误。  **【算法分析】**  这次实验我先是使用了递归的方法，对于问题进行分析，得出了f(n)=f(n-1)+f(n-2)\*2的结论，然后使用递推的方法来进行求解。同时使用了高精度计算，来防止数据过大计算后得不出最终答案。 | | | |