算法设计与分析第一次作业

计算机72班 姓名:徐子越 学号:2174410455

2-7:

原题多项式P(x) = (x-n1)(x-n2)……(x-nd);

如果d为偶数,根据分治法,可将d个乘式分组,每两个一组,可以分成d/2组,然后用题目给出的O(i\*logi)算法将每两组相乘,一轮下来得到d/2个新的分组,再使用上述算法得到d/4个分组,以此类推最后合并为一个即可.

如果d为奇数,则可以任意去出一个乘式,变成偶数情况,最后再乘上即可.

故,

T(d) = 2 \* T(d/2) + O(d\*logd) d>1

T(1) = O(1) d=1

解此递归式可以解得: T(d) = O(d\*log^2d)

2-8:

可以采用分治法解此题.

算法思路为,先取中位数对应的数组中的数,比较大小.

如果a[ i ] == i ,则得到结果

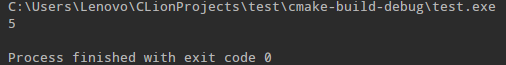
如果a[ i ] < i, 则取a[ i+1 ~ tail ]段进行同策略操作,取中位数比较

如果a[ i ] > I, 则取a[ head ~ i-1 ]同策略比较中位数操作

C++程序如下

#include <iostream>  
using namespace std;  
int find(int s, int e, int a[]){  
 int middle = (s+e)/2;  
 if (a[middle]==middle) return middle;  
 else if (s==e) return 0;  
 if (a[middle] > middle) return find(s, middle-1, a);  
 else if (a[middle] < middle) return find(middle+1, e, a);  
}  
int main(){  
 int test[] = {-3, -1, 0, 1, 3, 5, 7, 8, 9};  
 cout<<find(0,8,test)<<endl;  
 return 0;  
}

输出结果如下:



2-27:

1>根据题目,采用课本所给线性时间选择算法,可以以O(n)时间复杂度取得中位数.

2>将数组中每个数都与该中位数做差取绝对值,如果取绝对值时发生正负变换,则用新的数组append下来

3>继续采用线性时间选择,选择出最小的k个数,在加上中位数(遍历自定义数组决定出是否改变符号)

4>end