**实验7 11-12oj**

1. 实验的目的和要求

学习c++编程的原理和方法

1. 实验内容
2. 实验准备

C++中有关于指针的问题

分治算法

贪心算法

(二)实验项目

**A**

题目描述

设计一个给予分治法的函数max(int arr[], int size)，找出整型数组arr中的最大值。   
注：   
需要定义并调用函数

分析

由于要求使用分治算法，那么采用二分的思想搜索，将数组分为两部分，分别搜索最大的数，不断递归调用该函数，得到最大数。

通过指针，将数组作为函数的参数

源代码

#include<iostream>

using namespace std;

int search(int \*p,int num);

int main()

{

int n,a[100];

cin>>n;

for(int i=0;i<n;i++)cin>>a[i];

cout<<search(a,n);

return 0;

}

int search(int \*p,int num)

{

if(num==1)return(\*p);

else if(search(p,num/2)>search(p+num/2,num-num/2))return (search(p,num/2));

else return(search(p+num/2,num-num/2));

}

测试数据

Input

5

3 1 2 4 5

Output

5

B

题目描述

某单位在某个湖里举行潜水比赛，这是一个团体项目，每一支队伍由n人组成，要求所有队员从起点（A岸）潜水到终点（B岸）。在潜水时必须使用氧气瓶，但是每支队伍只有一个氧气瓶。最多两人同时使用一个氧气瓶，但此时两人必须同步游泳，因此两人到达终点的时间等于较慢的一个人单独游到终点所需的时间。大家都很友好，任何两个人愿意一起游泳。安排一种潜水的策略，使得最后一名队员尽量早到达终点。   
编写一个程序，首先输入某个队伍的人数n（n<100），接着依次输入n个队员的游到终点的时间，输出所有队员最早到达终点的时间。

分析

非常非常经典的一道贪心算法，一开始会以为只要一直用时最短的人和最长的人先一起过去，然后用时最短的回来，但是并不是这样的，考虑一种状态如何将最慢的两个人送到河对岸并且用时最短，只要每次都保证一次状态变换时间最短就可以了，那么首先对数据进行一次快速排序，可以得出，T1<T2….<Tn-1<Tn,那么有如下两种策略，

1. （1，6）（1）（1，5）（1），使得最后两人在对岸
2. （1，2）(1) (5 6)(2) 同样是最慢两人

时间分别为 T6+T1+T5+T1和T2+T1+T6+T2

那么简单的选择时间短的就可以了

函数用指针实现数组调用

源代码

#include<iostream>

using namespace std;

void qsort(int,int);

int ff(int \*p,int num,int ans);

int n,a[1000];

int main()

{

cin>>n;

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

cin>>a[i];

qsort(0,n-1);

cout<<ff(a,n,0);

return 0;

}

void qsort(int l,int r)

{

int mid=a[(l+r)/2],tmp,i=l,j=r;

do{

while(a[i]<mid)i++;

while(a[j]>mid)j--;

if(i<=j)

{

tmp=a[i];

a[i]=a[j];

a[j]=tmp;

i++;

j--;

}

}while(i<=j);

if(i<r)qsort(i,r);

if(j>l)qsort(l,j);

}

int ff(int \*p,int num,int ans)

{

if(num==0)return(ans);

else if(num==1)return(ans+\*p);

else if(num==2)return(ans+\*(p+1));

else if(num==3)return(ans+\*p+\*(p+1)+\*(p+2));

else

{

if(\*p+\*(p+num-2)>2\*\*(p+1))return(ff(p,num-2,ans+2\*\*(p+1)+\*p+\*(p+num-1)));

else return(ff(p,num-2,ans+2\*\*p+\*(p+num-1)+\*(p+num-2)));

}

}

测试数据

Input

4

1 8 2 5

Output

15

Input

10

1 2 3 5 8 9 11 15 20 1

Output

55

C输入一个字符串str， 再输入一个字符c，将字符串str中出现的所有字符c删除。要求定义并调用函数delch(str, c)，它的功能是将字符串str中出现的所有字符c删除

分析

由于str的长度不确定，有两种办法，使用动态数组，或者使用stdio.h中的gets函数，那么我才用了后者，接下来调用函数，通过指针实现数组作为参数的传递，然后对于字符型数组遍历，只要有C地方二重循环将之后的字符覆盖上来即可。

源代码

#include<iostream>

#include<stdio.h>

using namespace std;

int main()

{char c;

char str[80];

void delchar(char \*str, char c);

gets(str);

cin>>c;

delchar(str, c);

cout<<(str);}

void delchar(char \*str, char c)

{int i, j;i = j = 0;

while(str[i] != '\0')

{if(str[i] != c){str[j] = str[i];j++;}i++;}str[j] = '\0';}

Input

eiwajhfilawuebliguaw w

w

Output

Eiajhfilauebligua

D

题目描述

在数组中查找特定元素是否存在。输入一个正整数n(1<n<=20),然后再输入n个整数存入数组arr中；接着输入一个整数key，在数组arr中查找key是否存在。如果在arr中找到了key，则输出相应的下标；否则输出“Not Found”。要求定义并调用函数search(arr, n, key)，它的功能是在数组arr中查找元素key，若找到则返回相应的下标，否则返回-1.

分析

简单的用指针传递数组的问题

源代码

#include<iostream>

using namespace std;

int search(int \*p,int n,int key);

int main()

{

int n;

cin>>n;

int \*arr=new int[n];

for(int i=0;i<n;i++)cin>>arr[i];

int key;

cin>>key;

if(search(arr,n,key)!=-1)cout<<search(arr,n,key);

else cout<<"Not Found";

delete []arr;

return 0;

}

int search(int \*p,int n,int key)

{

for(int i=0;i<=n-1;i++)

if(p[i]==key)return(i);

return(-1);

}

测试数据

Input

4

3 7 8 2

8

Output

2

Input

5

1 2 3 4 5

6

Output

Not Found

1. 小结

本次实验着重于两个地方一着重于练习了指针的使用方法，特别是如何建立指针和一维数组之间的联系，（二级指针于二维数组联系目前没涉及），通过指针来在函数之间传递数组从而解决问题，四道题无一例外的都是用到了。

最后一道题还使用了动态数组，通过指针，new, delete来实现动态数组的创建，使用，调用和最后千万不能忘记的消亡

二则是练习了两个很重要的算法分治算法和贪心方法，分治就是把问题不断地缩小缩小最后通过递归来实现整合。贪心则是在于第二题，如何保证每一步的决策都是利益最大化的，并且我这种利益最大化的操作一定是建立在之后的操作一定成立保持总体最大化的一种思想，通过递归建立联系。（与之对应还有两外一种动态规划的算法，更加有趣。）

总体实验有一定的难度，可以在指针使用和贪心算法上再联系，下次实验可以比较贪心和动态规划两种相似有不同的思想，