华南理工大学

《算法设计与分析》课程实验报告

实验题目： 输油管道问题

姓名： 张锦怡 学号： 2016668555625

班级： 16信息安全班 组别：

合作者：

指导教师： 何克晶

|  |
| --- |
| **实验概述** |
| 【实验目的及要求】  实验目的：  练习快速排序算法的写法  实验要求：  某石油公司计划建造一条由东向西的主输油管道。该管道要穿过一个有n 口油井的油田。从每口油井都要有一条输油管道沿最短路经(或南或北)与主管道相连。如果给定n口油井的位置,即它们的x 坐标（东西向）和y 坐标（南北向）,应如何确定主管道的最优位置, 即使各油井到主管道之间的输油管道长度总和最小的位置?证明可在线性时间内确定主管道的最优位置。 给定n 口油井的位置,计算各油井到主管道之间的输油管道最小长度总和。要求使用快速排序.  【实验环境】  操作系统：  Win 10 |
| **实验内容** |
| 【实验方案设计】  思路分析：  若只有三口油井时，当其余两个分布在中间一口油井两边时，此时各油井到主管道之间的输油管道长度最小  则可以想象，若有多口油井，则根据某一个方向上选择排列最中间的油井的位置铺设管道，则是最小的。  【实验过程】（实验步骤、记录、数据、分析）  （1）码快速排序的实现  void swap(int a[], int i, int j)//数组交换  {  int t;  t = a[i];  a[i] = a[j];  a[j] = t;  }  int partition(int a[], int left, int right)//把数组分成有序的两部分，左边一律小于右边  {  int l = left;  int r = right;  int middle = (l + r) / 2;  int pivot = a[middle];  while (l <= r)  {  while (a[l]<pivot)//当左边大于选择的轴值跳出循环  {  l++;  }  while (a[r]>pivot)//当右边小于选择的轴值就跳出循环  {  r--;  }  if (l <= r)  {  swap(a, l, r);//两者进行交换  l++;  r--;  }  }  return l;  }  void quicksort(int a[], int left, int right)  {  int index = partition(a, left, right);  if (left<index - 1)  {  quicksort(a, left, index - 1);//对左边进行快排  }  if (index<right)  {  quicksort(a, index, right);//对右边进行快排  }  }  相关注释已在代码中交代清楚。  （2）根据所给的输入输出格式进行设置main函数，然后在以特定格式输出  （3）测试结果：    实验结果测试数据一    测试数据二 |
| **小结** |
| **一道看似复杂的题，就这么用快速排序解决了，所以感觉这种从小情况推到大情况的方法很管用，思路也很清晰，快排的代码实现的也感觉非常巧妙，以及这种算法的高效性。** |
| **指导教师评语及成绩** |
| 评语：  成绩：           指导教师签名：                                                 批阅日期： |

# 附录一

代码：

#include<iostream>

using namespace std;

void swap(int a[], int i, int j)//数组交换

{

int t;

t = a[i];

a[i] = a[j];

a[j] = t;

}

int partition(int a[], int left, int right)//把数组分成有序的两部分，左边一律小于右边

{

int l = left;

int r = right;

int middle = (l + r) / 2;

int pivot = a[middle];

while (l <= r)

{

while (a[l]<pivot)//当左边大于选择的轴值跳出循环

{

l++;

}

while (a[r]>pivot)//当右边小于选择的轴值就跳出循环

{

r--;

}

if (l <= r)

{

swap(a, l, r);//两者进行交换

l++;

r--;

}

}

return l;

}

void quicksort(int a[], int left, int right)

{

int index = partition(a, left, right);

if (left<index - 1)

{

quicksort(a, left, index - 1);//对左边进行快排

}

if (index<right)

{

quicksort(a, index, right);//对右边进行快排

}

}

int main()

{

int n;

cin >> n;

int \*x = new int[n];

int \*y = new int[n];

int i = 0;

int j = 0;

while (n--)

{

int a, b;

cin >> a >> b;

x[i++] = a;

y[j++] = b;

}

quicksort(y, 0, j - 1);

int result = y[j / 2];

int sum = 0;

for (int k = 0; k<j; k++)

{

sum += abs(result - y[k]);

}

cout << sum << endl;

}