算法分析与设计第二次作业

57119134 黄浩

2021 年 8 月 28 日

1 题目一

求下列递推关系表示的算法复杂度

1.1 $T(n) = 9T(n/3) + n$

解:

$\because n^{log_3 9} = n^2 > n$

$\therefore$ 根据主方法结论, 算法复杂度为:$O(n^2)$

1.2 $T(n) = n + 3T(n/4)$

解: $\because n^{log_4 3} < n$

$\therefore$ 根据主方法结论, 算法复杂度为:$O(n)$

1.3 $T(n) = 4T(n/2) + n^2 lgn$

解:

由于 $n^{log_2 4} = n^2$ 且 n^2 没有在多项式意义上大于或者小于 $n^2 lgn$

所以我们根据递归展开进行计算:

$\because$ 总共会递归 $log_2 n$ 层

$\therefore$ 最后一层叶子节点的计算复杂度为:

$$4^{log_2 n} = n^2 \quad [1]$$

$\because$ 第 x 层的计算复杂度为 $4^x(\frac{n}{2^x})^2 lg\frac{n}{2^x} = n^2 lg\frac{n}{2^x}$

∴ 第一层到最后一层的累加复杂度为:

$$\sum_{i=0}^{log_2 n-1} n^2 lg\frac{n}{2^i} = n^2 lg\frac{n^{log_2 n}}{2^{\sum_1^{log_2 n-1} i}} = n^2 lg\frac{n^{log_2 n}}{2^{\frac{log_2 n(log_2 n-1)}{2}}} = n^2 lg\frac{n^{log_2 n}}{n^{\frac{log_2 n-1}{2}}} \quad [2]$$

综上所述，算法总复杂度为 [1] + [2]:

$$n^2 + n^2 lg\frac{n^{log_2 n}}{n^{\frac{log_2 n-1}{2}}} = O(n^2(lgn)^2)$$

2 题目二

假设谷歌公司在过去 n 天中的股票价格记录在数组 A[1..n] 中，我们希望从中找出两天的价格，其价格增幅最大，即找到 A[i] 和 A[j] ($i \leq j$) 使得 M = A[j] − A[i] 的值最大，请设计一个时间复杂度不超过 O(n lg n) 的分治算法

答:

我设计了 int stockMaximum(A[l..r], int& min, int& max, int& ans) 算法用于在 l 到 r 段上进行计算，其中 min 存储这一段的最小值，max 存储这一段的最大值，ans 存储题目在这一段上的答案。算法伪代码如下:

Algorithm stockMaximum

1: **if** $l == r$ **then**
2: $min = A[l], max = A[l], ans = 0$
3: **else**
4: $mid = (l + r)/2$
5: $stockMaximum(A[l..mid], lmin, lmax, lans)$//计算左边的情况
6: $stockMaximum(A[mid + 1..r], rmin, rmax, rans)$//计算右边的情况
7: $min = Min(lmin, rmin)$//Min 表示求最小值
8: $max = Max(lmax, rmax)$//Max 表示求最大值
9: $ans = Max(lans, rans, rmax - lmin)$
10: **end if**

上述算法通过分治将问题分成三种情况:

1. 最优买卖解完全发生在数组左边 → 直接递归

2. 最优买卖解完全发生在数组右边 → 直接递归

3. 最优买卖解中的买行为发生在数组左边，卖行为发生在数组右边 → 用右边最大值减去左边最小值

我们可以通过主方法求得上述算法的复杂度：

$T(n) = 2T(n/2) + O(1)$

$\because n^{log_2 2} = n > 1$

$\therefore$ 根据主方法结论，算法复杂度为:$O(n)$

所以我们达到了题目要求。

3 题目三

问题描述：在与联盟的战斗中连续失败后，帝国撤退到最后一个据点。根据其强大的防御系统，帝国击退了联盟攻击的六波浪潮。经过几个不眠之夜，联盟将军亚瑟注意到防御系统的唯一弱点就是能源供应。该系统由 N 个核电站供电，其中任何一个都会使系统失效。

这位将军很快就派 N 名特工进行突击，这些特工进入了据点。不幸的是，由于帝国空军的袭击，他们未能降落在预期位置。作为一名经验丰富的将军，亚瑟很快意识到他需要重新安排计划。他现在要知道的第一件事是哪个特工离任何一个核电站最近。你是否可以帮助将军计算特工与核电站之间的最小距离?

答:

首先我设计了一个 struct 用于存储题目中的所有节点，其中除了 x、y 坐标信息外还有一个表示节点是核电站坐标还是特工坐标的一个 flag 信息；接着我设计了 shortestDistance(int left, int right, Node[1..2N], int&ans) 算法用于进行某一区间内特工与核电站之间的最小距离的计算，其中 left 存储这一区间的左边界 (Node 的下标)，right 存储这一区间的右边界 (Node 的下标)，Node 存储已经根据 x 坐标优先排好序的描述所有核电站和特工信息的 struct，ans 存储题目在这一区间上的答案。算法伪代码在最后一页 (注意在一开始已经对 Node 进行了预处理，即默认 Node 已经是按照 x 坐标优先的规则进行了排序)

我们的算法通过分治将问题分成三种情况：

1. 最近两点 (flag 不同) 在数组左边 → 直接递归

2. 最近两点 (flag 不同) 在数组右边 → 直接递归

3. 最近两点 (flag 不同) 一点在 mid 左侧 (包含 mid) 一点在 mid 右侧 → 直接计算横纵坐标之差都小于 d 的此两点距离

算法复杂度计算如下:

预处理排序复杂度为 $O(nlgn)$

分治递归式: $T(n) = 2T(n/2) + O(n)$

$\because n^{log_2 2} = n == n$

$\therefore$ 根据主方法结论, 算法复杂度为:$O(nlgn)$

所以结合预处理和分治的复杂度, 我的算法总复杂度为 $O(nlgn)$。

算法伪代码由于长度排版问题, 安排在了最后一页 (即下一页)。

Algorithm shortestDistance

if $l == r$ **then**
  $ans = INT\_MAX$//使用 INT_MAX 表示无穷大值
else
  $mid = (l + r)/2$
  $shortestDistance(left, mid, Node, lans)$//计算左边的情况
  $shortestDistance(mid + 1, right, Node, rans)$//计算右边的情况
  $d = Min(lans, rans)$//Min 表示求最小值
  for i = left to right **do**
    if $abs(Node[i].x - Node[mid].x) < d$ **then**
      if $abs(Node[i].y - Node[mid].y) < d$ **then**
        $midNode.push\_back(Node[i])$
        //与 mid 节点横纵坐标之差都小于 d 的点放入 midNode 中
      end if
    end if
  end for
  for i = 0 to midNode.size() **do**
    for j = i + 1 to midNode.size() **do**
      if $midNode[i].flag \neq midNode[j].flag$ **then**
        $d = Min(d, dis(midNode[i], midNode[j]))$//dis 计算两点距离
      end if
    end for
  end for
  $ans = d$//相当于返回答案
end if