Design and Analysis of Algorithms

1.计算下列T(n)的表达式：

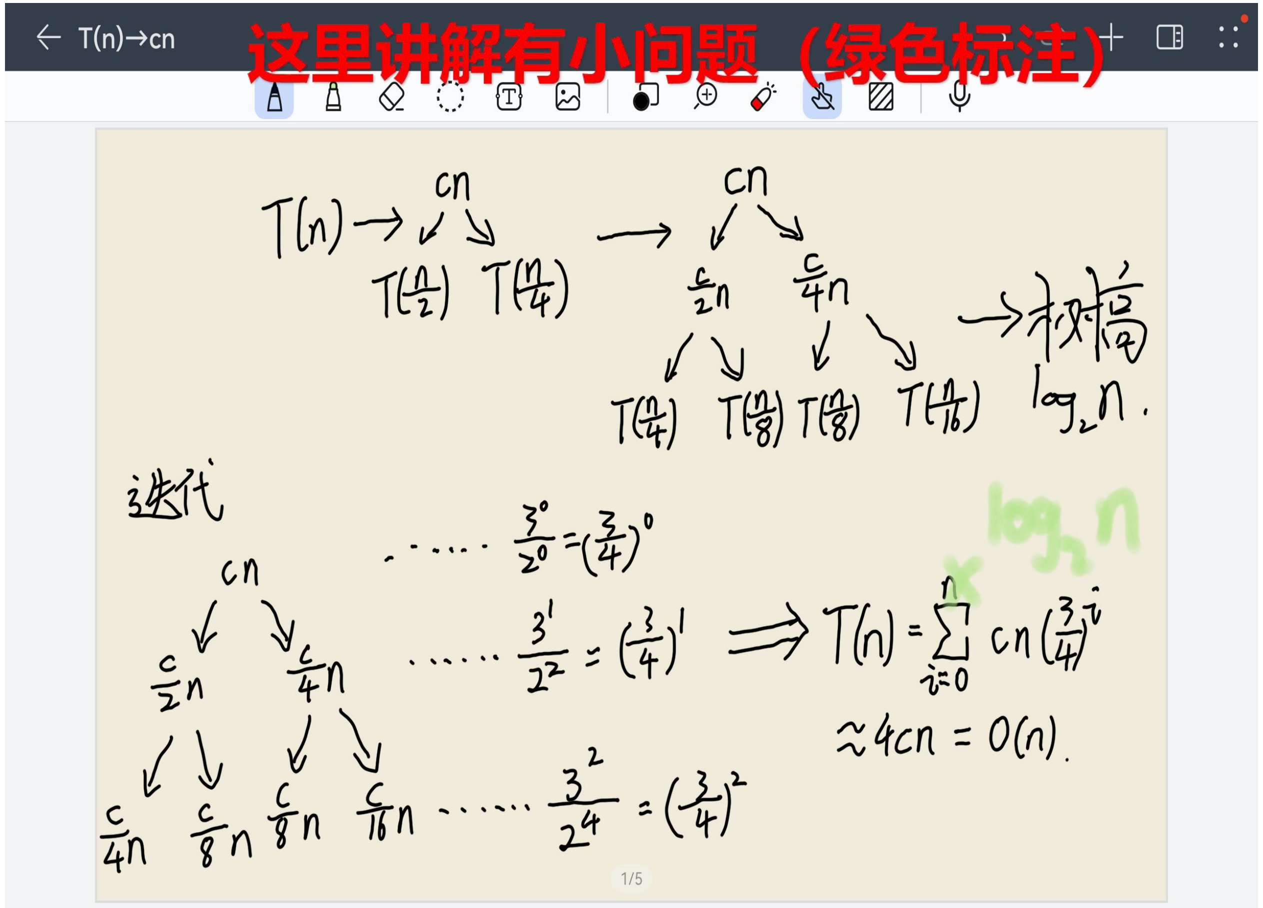
(1)$T(n) = T(n - 1) + n^2, T(1) = 1$

(2)$T(n) = T(n / 2) + T(n / 4) + cn, T(1) = 1$

$$T(n) = T(n - 1) + n^2$$
$$T(n - 1) = T(n - 2) + (n - 1)^2$$
$$T(n - 2) = T(n - 3) + (n - 2)^2$$

......

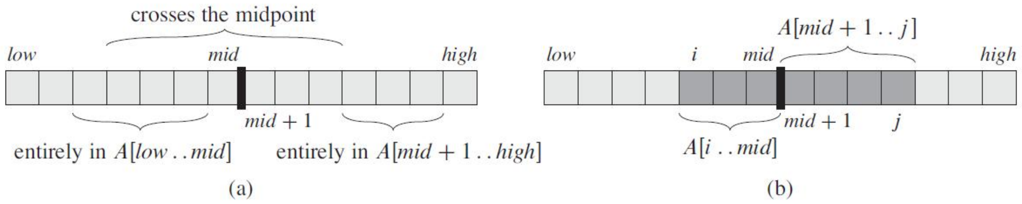
$$T(n) = n^2 + (n - 1)^2 + (n - 2)^2 + \dots\dots + 1$$
$$= n * (n + 1) * (2n + 1) / 6$$



Design and Analysis of Algorithms

2.给你一个整数数组nums，请你找出一个具有最大和的连续子数组（子数组最少包含一个元素），返回其最大和。（子数组是数组中的一个连续部分）

1.求一个数组中的最大子数组，利用分治算法，首先第一步需要做的就是将该数组划分为两个规模相当的子数组
2.第二步就是要分别考虑求解两个子数组的最大子数组，但是也会出现另外一种情况:原数组的最大子数组可能起始位置位于mid左边，终止位置位于mid右边，即cross_mid,
3.但无论如何，具体情况只可能是以下三种:
（1）最大子数组位于mid左边
（2）最大子数组位于mid右边
（3）cross_mid
可以维护四个变量
lsum表示[l,r]内以l为左端点的最大子段和
rsum表示[l,r]内以r为右端点的最大子段和
msum表示[l,r]内最大子段和
isum表示[l,r]的区间和




```
struct Status {
    int lSum, rSum, mSum, iSum;
};

Status pushUp(Status l, Status r) {
    int iSum = l.iSum + r.iSum;
    int lSum = max(l.lSum, l.iSum + r.lSum);
    int rSum = max(r.rSum, r.iSum + l.rSum);
    int mSum = max(max(l.mSum, r.mSum), l.rSum + r.lSum);
    return (Status) {lSum, rSum, mSum, iSum};
};

Status get(vector<int> &a, int l, int r) {
    if (l == r) {
        return (Status) {a[l], a[l], a[l], a[l]};
    }
    int m = (l + r) >> 1;
    Status lSub = get(a, l, m);
    Status rSub = get(a, m + 1, r);
    return pushUp(lSub, rSub);
}
```

Design and Analysis of Algorithms

3.有n个题目，每个题目有mi的做题时间和vi的分数，问得到V分数的前提下，最少需要多少时间做题（假设题目全对），请写出动规方程，算法思路和伪代码。

设Test(i, v)是用i个题目得v分数所要用到的时间，最终Test(n, V)是我们所求的解。

$$Test(i, v) = \begin{cases} 0 & i=0 \\ Test(i-1, v) & v_i > V \\ \min\{Test(i-1, v), Test(i-1, V-v_i) + m_i\} & \text{其它} \end{cases}$$

算法思路：看第i道题目的分值，如果比所需分值还要大就不做这道题。如果vi比所需V要小就重新考虑不做这道题所需时间与做这道题所需时间哪个更小。

伪代码：Test(n, V)

```
建立一个n×V的表格K
for i = 1 to n
    k[i,0] = 0
for j = 1 to V
    K[0,j] = 0
for i = 1 to n
    for j = 1 to V
        if j < vi
            k[i,j] = k[i-1, j]
        else
            k[i,j] = min{k[i-1, j], k[i-1, j-vi] + mi}
```

n \ V	0	1	2	3	4
0	0	0	0	0	0
1	0				
2	0				
3	0				
4	0				

Design and Analysis of Algorithms

4.Describe an efficient algorithm that, given a set { x1 , x2 , ..., xn } of points on the real line, determines the smallest set of unit-length closed intervals that contains all of the given points. Argue that your algorithm is correct. （16.2-5）

算法步骤:

1.排序:将所有点x1,x2,.. . , xn按照位置从小到大排序。

2.初始化:设置一个空集合Ｉ来存放所需的单位长度闭区间。

3.遍历:对排序后的每个点xi，如果xi已经被覆盖，则继续检查下一个点;如果xi未被覆盖，执行第4步。

4.放置区间:在点xi的位置放置一个新的单位长度闭区间[xi, xi＋1]，并将此区间加入到集合Ｉ中。

5.重复:重复步骤3，直到所有点都被覆盖。

Design and Analysis of Algorithms

4.Describe an efficient algorithm that, given a set { x1 , x2 , …, xn } of points on the real line, determines the smallest set of unit-length closed intervals that contains all of the given points. Argue that your algorithm is correct. （16.2-5）

正确性论证:
贪心选择性质:当我们遇到一个未被覆盖的点xi时，选择放置区间[xi, xi + 1]是有效的，因为这个区间覆盖了从xi开始的尽可能多的点，同时尽可能推迟了下一个区间的放置，从而减少了总区间的数量。
最优子结构:当我们放置了一个新的区间后，剩余未被覆盖的点的集合仍然是一个同样的问题，但规模减小了，这意味着我们可以递归地应用同样的策略来解决。
算法效率:排序点的时间复杂度为O(n log n)，遍历点并放置区间的时间复杂度为O(n)。因此，整个算法的时间复杂度为O(n logn)，这是非常高效的。