1、统计数组逆序对，时间复杂度 O(nlogn)
归并排序，如果右边的向量小，sum+=左边向量的长度

2、判断数组中是否有和为 s 的两个数，时间复杂度 O(nlogn)

```
Use Merge Sort to sort the array A in time Θ(n lg(n))
i = 1
j = n
while i < j do
    if A[j] + A[j] = S then
        return true
    end if
    if A[i] + A[j] < S then
        i = i + 1
    end if
    if A[i] + A[j] > S then
        j = j − 1
    end if
end while
return false
```

3、最长单增子序列

设计一个 $O(n\lg n)$ 时间的算法，求一个 n 个数的序列的最长单调递增子序列。（提示：注意到，一个长度为 i 的候选子序列的尾元素至少不比一个长度为 $i-1$ 候选子序列的尾元素小。因此，可以在输入序列中将候选子序列链接起来。）

Algorithm 6 LONG-MONOTONIC(S)

```
Initialize an array B of integers length of n, where every value is set equal
  to ∞.
Initialize an array C of empty lists length n.
L = 1
for i = 1 to n do
    if A[i] < B[1] then
        B[1] = A[i]
        C[1].head.key = A[i]
    else
        Let j be the largest index of B such that B[j] < A[i]
        B[j + 1] = A[i]
        C[j + 1] = C[j]
        C[j + 1].insert(A[i])
        if j + 1 > L then
            L = L + 1
        end if
    end if
end for
Print C[L]
```

4、判断众数（个数大于 n/2 的）

先走一趟，不一样的删掉，剩下的那个数可能是，然后再走一趟验证一下

5、

我们将一棵树 $T=(V, E)$ 的直径定义为 $\max_{u,v\in V}\delta(u, v)$，也就是说，树中所有最短路径距离的最大值即为树的直径。请给出一个有效算法来计算树的直径，并分析算法的运行时间。

两次 DFS，第一次 DFS 最后的那个点是 U，那么拿着 U 再做一次 DFS

6、

（有向无环图中的最长简单路径） 给定一个有向无环图 $G=(V, E)$，边权重为实数，给定图中两个顶点 s 和 t。设计动态规划算法，求从 s 到 t 的最长加权简单路径。子问题图是怎样的？算法的效率如何？

Problem 15-1

Since any longest simple path must start by going through some edge out of s, and thereafter cannot pass through s because it must be simple, that is,

$$LONGEST(G, s, t) = 1 + \max_{s\sim s'}\{LONGEST(G|_{V\setminus\{s\}}, s', t)\}$$

with the base case that if $s = t$ then we have a length of 0.

A naive bound would be to say that since the graph we are considering is a subset of the vertices, and the other two arguments to the substructure are distinguished vertices, then, the runtime will be $O(|V|^2 2^{|V|})$. We can see that we can actually will have to consider this many possible subproblems by taking $|G|$ to be the complete graph on $|V|$ vertices.

7、

Borden 教授提出了一个新的分治算法来计算最小生成树。该算法的原理如下：给定图 $G=(V, E)$，将 V 划分为两个集合 V_1 和 V_2，使得 $|V_1|$ 和 $|V_2|$ 的差最多为 1。设 E_1 为端点全部在 V_1 中的边的集合，E_2 为端点全部在 V_2 中的边的集合。我们递归地解决两个子图 $G_1=(V_1, E_1)$ 和 $G_2=(V_2, E_2)$ 的最小生成树问题。最后，在边集合 E 中选择横跨切割 V_1 和 V_2 的最小权重的边来将求出的两棵最小生成树连接起来，从而形成一棵最后的最小生成树。

请证明该算法能正确计算出一棵最小生成树，或者举出反例来明说该算法不正确。

Exercise 23.2-8

Professor Borden is mistaken. Consider the graph with 4 vertices: a, b, c, and d. Let the edges be $(a, b), (b, c), (c, d), (d, a)$ with weights 1, 5, 1, and 5 respectively. Let $V_1 = \{a, d\}$ and $V_2 = \{b, c\}$. Then there is only one edge incident on each of these, so the trees we must take on V_1 and V_2 consist of precisely the edges (a, d) and (b, c), for a total weight of 10. With the addition of the weight 1 edge that connects them, we get weight 11. However, an MST would use the two weight 1 edges and only one of the weight 5 edges, for a total weight of 7.