**贪心 二分 前缀和 差分 双指针**

贪心

“贪心算法（greedy algorithm，又称贪婪算法）是指，在对问题求解时，总是做出在当前看来是最好的选择。也就是说，不从整体最优上加以考虑，算法得到的是在某种意义上的局部最优解。”

贪心的本质是选择每一阶段的局部最优，从而达到全局最优。

贪心算法没有固定的框架，算法设计的关键是贪婪策略的选择。

反向贪心：最后的条件对结果起决定作用时

反悔贪心：搭配优先队列，可以返回修改

使用贪心算法的一般步骤

有点与动态规划类似，这就导致了我们有些题很难想出来是应该用贪心还是动态规划。

• 将问题划分为若干个子问题

• 找出适合的贪心策略

• 求解每一个子问题的最优解

• 将局部最优解堆叠成全局最优解

前缀和和差分一般用于做预处理

差分，前缀和

二维差分：

定义sum[i][j]为a[1 ~ i][1 ~ j]的值总和

Sum[x1 ~ x2][y1 ~ y2] = sum[x2][y2] – sum[x2][y1 – 1] – sum[x1 – 1][y2] + sum[x1 – 1][y1 – 1]

二分

二分带不带等号由check决定

int l = 1,r = n;

int ans = -1;

while(l <= r)

{

int mid = (l + r) >> 1;

if(jud(mid))

{

ans = mid;

l = mid + 1; //l与r的改变方向待定

}

else

{

r = mid - 1;

}

}

双指针

就是同时使用两个指针，在序列、链表结构上指向的是位置，在树、图结构中指向的是节点，通过或同向移动，或相向移动来维护、统计信息。