更正

原来A的题解中 把1换成10应该改为把1换成01。

A

如果两个操作得到的串相同，那么我们可以认为这两个是一样的。

所以我们可以把问题转化成：

- 在开头插入一个1。
- 把一个0换成01。
- 把一个1换成01。
- 在末尾插入一个0。

容易发现，这样就能保证不重不漏。假设在开头补一个0，结尾补一个1，那么可以看成每次可以选择一个01删除其中的一个0或者1。我们在每个01之间插入分隔符。比如`101->0A1B0C1D1`，每次选择其中一个01删除的时候，我们同时也把分隔符删除。比如`0A1B0C1D1 -> 0A1B0D1->0B0D1->0B1`。那么每个分割符被删除的时间相当于一个$0\sim n$的排列，如果s_i元素是0，那么右边的分隔符会先删除，否则左边的分隔符先删除。这个条件是充要的。

所以问题等价于把01换成`<`和`>`，求满足条件的排列个数。这个可以用容斥原理+分治FFT进行计算。时间复杂度$O(n\log^2 n)$。

B

首先如果指定了顾客，那么一定是按b逆序服务。

然后考虑怎么dp，顺着做比较困难，考虑到着做，也就是假设顾客全都要在0时刻吃完，那么第i个菜需要在$-b_i$时刻之前开始做，也就是开始时间要在$b_i+a_1+a_2+\cdots+a_i$之前，答案就是所有的i的最大值。可以这么计算：

```
ans = 0
for i = n ... 1
  ans = max(ans, b[i]) + a[i]
```

所以可以用$O(n^2)$ dp解决，也就是前i个物品选了j个，这个ans的最小值是多少，但是这个显然不能改成计数。具体是这样：$dp_{i,j}=\min(dp_{i-1,j},\max(dp_{i-1,j-1},b_i)+a_i)$

考虑怎么维护这个dp数组，可以发现，一旦$dp_{i-1,j}\le b_i$，那么$dp_{*,j}$这一项不会再发生变化。考虑维护一下现在可能变化的dp值的差分数组。也就是令dp_{i-1,j_0}为第一项大于b_i的元素，那么定义差分数组f_j，当$j>j_0$时$f_j=dp_{i-1,j}-dp_{i-1,j-1}$。当$j=j_0$时，$f_j=dp_{i-1,j}-b_i$。那么这个转移相当于在$f$数组中插入了一项$a_i$。并且可以归纳证明$f$是递增的，也就是所有可能变化的dp值一定是凸的。

所以可以用一个优先队列取维护f数组，每次加入一个元素i，令$d=b_i-b_{i-1}$，那么先减少队列头的元素，如果减到了0就弹出，直到减少的总量等于d。弹出的元素就是确定的dp值。然后在队列内加入a_i。当我们弹出到k个元素之后，减少的总量就是k的答案。

然后考虑怎么计数。如果我们把整个优先队列压下来，那么状态有点大。考虑线性性，也就是每次将加入a_i的时候，考虑这个a_i会在哪个位置弹出。

那么在a_i加入之后，如果知道了已经在a_i之前有多少个数(包括在队列里面的和已经pop的)，以及还在队列里面的在a_i之前的元素的和，那么就可以直接dp了。

对于a_i之前的元素更加困难一点。做法是每次加入一个元素就去定下这个元素应该在a_i之前出队列还是之后。那么我们需要记录一下确定在a_i之前的有多少个，当前还在队列里的a_i之前的元素之和，当前还在队列里的a_i之前的元素最大值，确定在a_i之后的元素的最小值，这样就可以dp了。

时间复杂度$O(n^3V^4)$。

C

考虑dp，令$dp_{u,j}$表示从u这个子树，延伸出来的链的长度为j的最大权值和。

考虑如何合并各个儿子的信息。如果$k=3$，相当于将一些长度为1和2的匹配起来，然后选择至多一条长度等于1或者2的链延伸上去。如果$k=4$，相当于将一些长度为1和3的匹配，长度为2的两两匹配，然后选择至多一条链延伸上去。

以$k=4$为例，如果我们直接做的话，合并各个儿子的时候，我们需要记录长度为2的链的个数的奇偶性，和当前的选择里面1比3多了多少。最后的答案就是2的个数为偶数，1和3的个数差不超过1，或者2的个数为奇数，1和3的个数相同。这么直接做的时间复杂度是$O(n^2)$的，不一定能通过。

这个问题，如果$k=3$，好像可以用某种贪心解决，但是$k=4$，因为2的存在好像不太好贪心。

考虑这个结果，一个随机的$1,-1$的数列，那么很大概率它的前缀和的绝对值是不会超过$O(\sqrt{n})$的。也就是意味着如果我们把一个节点的儿子随机排列，那么对于最优解，很高概率，合并的时候1和3的个数差不会超过$O(\sqrt{n})$，也就是做合并儿子的dp的时候这一维不需要记录超过$O(\sqrt{n})$的值。

所以整个时间复杂度是$O(n\sqrt{n})$的。实践过程中取1000就够了。