A

由于优惠券可以重复使用，因此使用情况仅取决于最靠前的物品

记rk_i为c中第i小的位置，D_i为第i个物品所能使用优惠券的d_j之和

设$t=\min_{1\le i<k} rk_i$，c_i最小的$k-1$个物品必然选择，并分类讨论：

- 不更新t，即选c_i第k小的物品，答案为$\sum_{i=1}^{k} c_{rk_i}-D_t$
- 更新t，答案为$\sum_{i=1}^{k-1} c_{rk_i}+\min_{1\le i<t}(c_i-D_i)$

时间复杂度为$O(n\log n)$

B

确定t后，设t最长的Border为$t[1,l]=t(m-l,m]$，形式即$t+t(l,m]+t(l,m]+\dots$

设s中字符c出现a_c个，t中出现b_c个，$t(l,m]$中出现d_c个，答案即$\min\lfloor\frac{a_c-b_c}{d_c}\rfloor+1$

确定l后，限制即$s_i=s_{i+(m-l)}$，从而划分成$m-l$个等价类，且$t(l,m]$恰遍历所有等价类

具体的，设$m=k(m-l)+r$，即有r个大小为$k+1$的等价类和$m-l-r$个大小为k的等价类

设$b_c=x_ck+y_c(k+1)$，则$d_c=x_c+y_c$且总可以取$m-l=\sum(x_c+y_c)$

换言之，问题即最小化x_c+y_c，也即要求$y_c\equiv b_c \pmod k$且尽量大

由于$k=\lfloor\frac{m}{m-l}\rfloor$仅有$O(\sqrt{m})$种取值，时间复杂度为$O(n+m+|\Sigma|\sqrt{m})$（其中$\Sigma$为字符集）

C

注意到一个测试点重复出现不影响结果

每个人有以下5种状态：时间/空间是/否达到最大值、是否已经出现非AC状态

对此状压DP，转移用bfs实现，时间复杂度为$O(nm5^n)$

D

对序列分治，并处理跨过当前分治中心的询问

问题可以看作对所有长为$2k$形如$0101\dots01$的子序列关于端点的一次函数乘积求和

对两侧分别DP，在状态中存储开头/结尾的字符及01的对数，并分别维护个数和端点之和

时间复杂度为$O(nk\log n+qk)$