A.

算法：思维+贪心（当然也可以二分）

题意：进行 k 次让任意一个数 $+1$，求最大的中位数。因为是只有 $+1$ 的操作，所以只存在中位数 t 和原本比 t 大的数进行交换位置。这里只需要考虑这个范围内的数就行了。可以用一种类似于填坑的思路来做：

题目要求数都尽量大，并且保证中位数一直都是中位。如果是这样的一个样例：

5 8

1 2 3 4 10

3 和 4 ， 3 可以和 4 齐平，用 1 次操作变成: 1 2 4 4 10，剩余操作次数 9 次。

而后两个 4 一起看，后一个数是 10 ，但是要齐平就需要 12 次操作，所以可以在 10 以内让这两个数尽量大，也就是再用 8 次操作变成：1 2 8 8 10，最大的中位数就是 8 了。还剩一次操作是没有意义的。

所以就可以得出结论：

如果可以和下一个数齐平，那就一起填上去；如果不够了，那就一起尽量变大，直到 k 个数用完即止。

B.

算法：数学

这道题的难点其实在于证明为什么拆出足够多的 33 就能使得乘积最大。证明如下：

（1）首先证明拆出的因子大于 4 是不行的。设 x 是一个因子，$x > 4$ ，那么可以将这个因子再拆成两个因子 $x-2$ 和 2，易证 $(x-2) \times 2 > x$。所以不能有大于 4 的因子。

（2）4 这个因子也是可有可无的，$4 = 2 + 2$，$4 = 2\ times 2$。因此 4 这个因子可以用两个 2 代替。

（3）除非没有别的因子可用，1 也不能选作因子。

（4）这样呢，就只剩下 2 和 3 这两个因子可以选了。下面再证明 3 比 2 好：

一个数 $x = 3m + 2n$，那么 $f = 3^m \times 2^n = 3^m \times 2^{\frac{x-3m}{2}}$ 可以对它取个对数。

$$
\begin{aligned}
lnf &= m \ln 3 + n \ln 2 \\
&= m \ln 3 + \frac{x-3m}{2} \ln 2 \\
&= \frac{x}{2} \ln 2 + (\ln 3 - \frac{3}{2} \ln 2) m
\end{aligned}
$$

其中 $\ln 3 - \frac{3}{2} \ln 2 > 0$ 所以 f 是 m 的增函数，也就是说 m 越大越好。所以 3 越多越好。

再多说一句，如果拆出的因子不限于整数的话，可以证明$e = 2.718\ldots$ 是最佳的选择。感兴趣的可以试着证明一下。

（5）但当一个数大于取了足够多的 3 的时候，最后还剩下一个 4 的时候，应该选择 4 作为因子，因为 $4 > 3 \times 1$!

E.

第一想法肯定是暴力搜索，但是发现复杂度为 2^{40} ，果断放弃。

但是正解确实是暴搜，只是需要换种姿势，2^{40} 确实搜不了，但是 2^{20} (大约 $1e6$) 可以搜，即折半搜索。

把输入的 n 个项目价格从中间分成两部分，对于每一部分分别进行复杂度为 2^{20} 的暴力搜索，分别存下两部分符合条件的总价。对第二部分的总价进行排序，然后遍历第一部分，对于每一个第一部分的总价，用 upper_bound或二分法在第二部分总价中找出能与之相加不超过 M 元的个数，计入答案。

复杂度为 $O(2^{n/2} \times \log 2^{n/2})$ 。

F.

题意：一个仅包含小写字母的字符串，每次可以有两种选择：

(1).结束游戏

(2).把一个字母变成它后面一个字母，如 $b \to c$ ，特别地 $z \to a$

Wiki想要使串的字典序尽可能大，ta的对手则向让其尽可能小

假设两人都绝顶聪明，Wiki先手，问结束游戏时的字符串长什么样

例：

$abc \to bbc$ ：Wiki把 a 变成 b ，对手结束游戏（如果对手不结束游戏，Wiki下一轮会把串变成 cbc ，会更大，所以对手只能选择马上结束游戏）

$yyyzaaa \to yyyzaaa$ ：Wiki直接结束游戏（Wiki如果吧 y 变成 z ，对手会把这个 z 变成 a ，就更小了；Wiki肯定不会自己把中间的 z 变成 a ；Wiki如果把z后面的 a 变成 b，对手会把 z 变成 a，就比一开始的串更小了，所以不行）

结论：除去前导 y 后的第一个字母如果是 z ，则Wiki直接终止游戏，原样输出；

否则Wiki选择将该字母 $+1$，然后对手选择结束游戏。

G.

普通平衡术。

双倍经验: https://www.luogu.org/problem/P3369

标程写的 $Treap$ ，⑧会的同学可以百度一下。

H.

双倍经验: https://www.luogu.org/problem/P2766

首先你可以暴力 dp 出原串的最长不下降子序列，然后大力最大流。

不会网络流的同学可以看看模板题 https://www.luogu.org/problem/P3376

考虑如下的建图方式：

首先，我们要保证每个元素只能被选择一次。经典拆点，把每个元素拆成两个点。对于 V_i ，所有的入边连向 V_{in_i} ，所有出边从 V_{out_i} 流出，然后从 V_{in_i} 往 V_{out_i} 连一条容量为 1 的边。

其次，我们要让每一条从源点到汇点的路径长度都恰好为 s 。

对于所有 $dp[i] == 1$ 的元素，源点向 V_{in_i} 连一条非零边。

对于所有 $dp[i] == s$ 的元素，V_{out_i} 向汇点连一条非零边。

对于第 i 个元素 a_i ，枚举元素 $j > i$ ，如果 $a_i \le a_j, dp[i] + 1 == dp[j]$ ，那么它是可以成为答案候补且不会破坏路径长度的，让 V_{out_i} 向 V_{in_j} 连一条非零边。

然后这张图的最大流就是答案啦。