1001

如果 $n \le d$，那么小 R 可以通过连续工作 n 小时直接完成，否则将进入一个周期为 $d+k$ 的工作循环，每个循环完成 d 小时的工作量，直到剩余工作量不超过 d 小时。因此答案为 $\lfloor \frac{(n-1)}{d} \rfloor * (d+k) + n - \lfloor \frac{(n-1)}{d} \rfloor * d$。时间复杂度为 $O(1)$。

此题数据范围较小，直接暴力模拟也可以通过。

1002

从左到右遍历的过程中，维护当前已经遍历过的 'a' 的数量，设其为 cnt。每次扫到 'k'，答案加上当前的 cnt 即可。时间复杂度为 $O(n)$。

1003

考虑二分答案，每次检查合法性时贪心地染色即可，注意特判答案为 0 时的情况。时间复杂度为 $O(n \log n)$。

1004

朴素的想法，我们可以用并查集维护连通性，每次询问我们将边权不大于 p_i 的边加入并查集中，同时维护每个连通块的最大点权，这样做的时间复杂度为 $O(qm\alpha(n))$。

我们可以将询问按 p_i 从小到大排序，处理询问 i 时先将边权不大于 p_i 且还没有处理过的边加入并查集中（将边按边权排序，用双指针维护已经处理过的边），同时维护每个联通块的最大点权，这样就不需要每次重构并查集了。总时间复杂度为 $O(m \log m + q \log q)$（上限在排序）。

本题还有另一种解法，建立 Kruskal 重构树，维护每个结点的子树最大点权 max_i，每次询问倍增跳到最高的合法点 p，答案取 max_p 即可。时间复杂度和第一种解法相同。

1005

我们可以很容易地想到一个 $O(nk)$ 的 dp做法：设 f_i 为只考虑前 i 个数的答案，我们可以枚举划分的最后一段的长度，状态转移方程为 $f_i = \min_{j=\max(0,i-k)}^{i-1}(f_j + \max(a_{j+1}, a_{j+2}, \ldots, a_i))$。

设当前枚举到 i，$s_j = f_j + \max(a_{j+1}, a_{j+2}, \ldots, a_i)$。考虑用数据结构维护 s_j，我们会发现，所有满足 $a_j \le a_i$ 的位置 j 都会被 a_i 覆盖到。因此只需要用单调栈维护一段 a_j 值递减的位置 j，每次枚举到一个新的 i 时，只需从栈顶依次弹出满足 $a_j \le a_i$ 的位置 j，并且在数据结构中区间更新待修正的 s_j 值。我们发现此数据结构需要支持区间加和区间最大值询问，因此我们可以用带 lazy-tag 的线段树完成这些操作。由于每个位置 j 最多进入一次单调栈，所以待更新的区间数量是 $O(n)$ 的，所以总时间复杂度为 $O(n \log n)$。

1006

本题其实比 1005 简单，可惜没人做。

考虑维护每个位置的 i 前缀异或和 s_i，我们可以发现题意等价于选取 $m+1$ 个位置（设第 i 个位置为 p_i，其中 $p_0 = 0$，$p_{m+1} = n$，$0 < p_1 < p_2 < \cdots < p_m < n$），使 $(s_{p_{m+1}} \bigoplus s_{p_m})\&(s_{p_m} \bigoplus s_{p_{m-1}})\&\ldots\&(s_{p_1} \bigoplus s_{p_0})$ 最大，其中 $\bigoplus$ 表示按位异或。

按位考虑，如果答案在二进制下某一位能取到 1，那么选出的这 $m+1$ 个位置在这一位必须 0、1 交错。因为要求最大值，所以我们需要优先满足高位。我们从高位到低位枚举，答案能在当前位取 1 当且仅当能选出**满足题意**的 $m+1$ 个位置，使这 $m+1$ 个位置的 s 值在当前位和已经确定取 1 的二进制位都满足 0、1 交错。时间复杂度为 $O(30n)$。

1007

本题来源于 CF1313D，解法比较复杂，有兴趣的可以去看一下原题。