T1

分类讨论：

- 两人无需求，切 0 刀。
- 一人无需求，另外两人需求相等，切 1 刀。
- 一人无需求，另外两人需求不等，切 2 刀。
- 三人均有需求，其中两人的需求之和等于另一人，切 2 刀。
- 三人均有需求，存在需求相等的两人，切 2 刀。
- 其余情况，切 3 刀。

T2

所有坐标排序后存入一个数组中，维护当前未出队的士兵对应的区间，初始为 $[1, n]$。

每次移动均累计入总的偏移量。对于一次移动，只需判断边上的的士兵是否会出队，并根据出队的情况调整区间边界即可。

因为每个士兵只会出队一次，所以时间复杂度为 $O(n+m)$。

T3

用栈模拟即可。

对于每次要入栈的一段连续整数，视为一个整体入栈，记录其左右边界 l, r。

出栈时，若栈顶元素对应的区间数字个数不超过 k，就可以直接出栈，对应的价值之和可以用求和公式 $O(1)$ 计算。若超过 k，则计算出会取出哪一段数字，同样用求和公式 $O(1)$ 计算，并修改栈顶元素对应的右边界。

时间复杂度 $O(n)$。

T4

如果 a 中某一金额能够被 a 中其他金额凑出来，则该金额可以移除，且不影响货币系统能表示出来的金额。

显然，决定一个金额 x 是否可以被移除的，是那些小于 x 的金额。

考虑对 a 中所有面额按从小到大排序，依次考虑每一个面额。

如果考虑了前 i 个面额之后，a_{i+1} 已经被表示出来了，则 a_{i+1} 可以移除。

如何处理前 i 种面额能表示出来的金额有哪些这一问题？完全背包。

设 f_i 表示金额 i 是否能被表示，有 $f_i| = f_{i-k\times a_j}$。

初始状态为 $f_0 = 1$。

时间复杂度 $O(T \times n \times MaxA)$。