E2

Problem: there are n books available, each of which is liked by Alice, Bob, both or neither. Each book has a cost. We have to choose exactly m books such that Alice likes at least k of the books, Bob likes at least k of the books, and the total cost is minimized.

Consider splitting the books into four piles; books that both like, books that only Alice likes, books that only Bob likes, and books that no one likes. Sort each pile by cost. Then a solution corresponds to selecting some prefix of each pile. Let the prefix lengths be p, q, r, s. We have to satisfy $p + q + r + s = m, p + q \geq k, p + r \geq k$. Each such quartet corresponds to a candidate solution, with an associated cost.

Suppose we fix some p_0 and ask for the minimum-cost candidate among all candidates with $p = p_0$ is. We know that we must select at least $q_0 = k - p_0$ books from the second pile, and similarly for the third pile. After these obligatory selections, we have selected $p_0 + 2(k - p_0)$ books, and this should not be larger than m. Let the difference be d. Then, the optimal selection is to select the cheapest d books from the merged suffixes of the second, third and fourth piles.

Hence it suffices to iterate through possible values of p and compute a candidate cost cheaply. One way to do this is in $O(\log n)$ time per iteration is to do parallel binary search on the suffixes (the starting point of the suffixes is easily computed). This is a similar binary search as the problem "median of two sorted arrays"). Total time for the iteration over p is $O(n \log n)$.

That approach considers each possible value of p independently, computing the candidate answer from scratch. Another approach is to iterate in decreasing order of p. Suppose we are at a solution p, q, r, s which is optimal (for the fixed value p) and we decrement p. It might be necessary to increment q, and it might be necessary to increment r. Suppose the new values after these optional decrements are p', q', r', s. Then the sum $p' + q' + r' + s'$ might be bigger or smaller than m, it differs from m by at most 2. If it is smaller, we can just pick some additional unpicked books from the 2-4th piles. If it is larger, we can drop books from the 4th pile, and maybe from the 2nd, and maybe from the 3rd (we cannot drop from the second pile if $q' = k$, for instance). In any case, each iteration computes a candidate from the previous candidate in $O(1)$ time.

A third approach is to iterate in increasing order of p, which does the inverse of the fixup operations from the second solution. An increment of p relaxes constraints on q and r, potentially allowing us to replace a selected book with a cheaper unselected one, and also requires selecting 1 more book overall.

After that, the new sum