

Мовні проблеми

Будемо перебирати відповідь за допомогою бінарного пошуку. Якщо ми можемо утворити пари використовуючи j волонтерів, то зможемо утворити якщо візьмемо $j + 1$ волонтерів. Як можна перевірити, що можна утворити пари, щоб виконувались $a_i + p_i \geq x$, де p_i — рівень знань волонтера у парі з i -ю людиною? Відсортуймо масив a за зростанням, і зробимо масив B , в який покладемо перші j волонтерів з масиву b . Відсортуємо масив B по спаданню та візьмемо перші n елементів. Тепер залишилось перевірити, що виконується $a_i + B_i \geq x$ для усіх i від 1 до n .

Доведемо, що це завжди буде оптимально. Припустимо, що ми маємо 4 індекси $i \leq k < t \leq j$, і дві пари (a_i, B_j) та (a_t, B_k) , для яких виконується $a_i + B_j \geq x$ та $a_t + B_k \geq x$. Доведемо, що також підходять пари (a_i, B_k) та (a_t, B_j) . За умовою маємо, що $a_i \leq a_t$ та $B_k \geq B_j$, розпишемо нерівності:

$$a_i + B_k \geq a_i + B_j \geq x$$

$$a_t + B_j \geq a_t + B_k \geq x$$

В обох нерівностях ми замінили більший елемент на менший для більшої сторони та отримали нерівність, яка правда. Отже, існує оптимальне розбиття на пари, для яких не існує пар (a_i, B_j) , (a_t, B_k) ($i \leq k < t \leq j$), а отже, парування (a_i, B_i) є одним з оптимальних.

Таке рішення матиме асимптотику $\mathcal{O}((n + m) \cdot \log^2 m)$, що недостатньо швидко, але при гарних оптимізаціях мало змогу пройти усі тести.

Подумаємо, як можна прибрати зайвий $\log m$ з асимптотики. Він з'являється з сортування масиву B . Як отримувати масив B вже відсортованим? Створимо вектор, який буде зберігати пари (b_i, i) . Посортуємо його по спаданню. Надалі ми зможемо створювати масив B ітеруючись по вектору і перевіряючи, чи належить цей елемент до тих, які ми розглядаємо, тобто $i \leq j$.

Це оптимізує рішення до асимптотики $\mathcal{O}((n + m) \cdot \log m)$, що достатньо, щоб не перевищити ліміт часу.