

Розбір задачі «Бульбашкове сортування»

Сортуємо всі масиви лексикографічно по спаданню, бо за кожен пробіг баблсорт масив стає лексикографічно меншим.

Автор задачі: Максим Зуб

Підготував: Максим Зуб

Розбір задачі «Система рівнянь»

Давайте знайдемо масив префіксних ксорів (*prefans*) відповіді (з нього легко зробити масив-відповідь). Нехай ми вже знайшли перші $i - 1$ елементів. Нехай $prefans_i = 0$ спочатку. Тоді маємо якісь відрізки, які закінчуються у позиції i . Рахуємо кількість рівнянь які зараз виповняються (k_0), та не виповняються (k_1). Якщо $k_1 > k_0$, то значення в $prefans_i$ змінюємо на 1.

Джерело: 1/8 ICPC у Польщі (Ягелонський університет)

Підготував: Максим Зуб

Розбір задачі «Бітова гра»

Будемо рахувати динаміку за 3^n , i -й біт рівен 0, якщо в масці був нуль і ми про нього не запитували, 1 якщо один і не запитували, і 2 якщо ми знаємо значення цього біта. Спочатку порахуємо SOS-динаміку $kol[mask]$ для масок, підмаскою маски є всі маски тільки з нулів та одиниць, які підходять під шаблон цієї маски (тобто якщо маска-шаблон має нуль або одиницю, то у підходящій маски має бути такий самий біт, якщо маска-шаблон має двійку, то підходяща може мати і нуль, і одиницю). Далі рахуватимемо динаміку для масок, де $dp[mask]$ - мінімальна кількість запитань, щоб дізнатися всі маски, що підходять під шаблон. $dp[mask] = 0$, якщо $kol[mask] \leq 1$. якщо j -й біт двійка, то $dp[mask] = \min(dp[mask], \max(dp[mask - 3^j * 2], dp[mask - 3^j]) + 1)$;

Джерело: 1/8 ICPC у Польщі (Ягелонський університет)

Підготував: Максим Зуб

Розбір задачі «Бойовий парад»

Якщо кількість доступних місць менша за $2 * k$ тоді відповідь 0. Нехай $R(j)$ буде числом розсадок таких що j пар сидять поруч. Тоді відповідь це $(\prod_{j=0}^{k-1} (\binom{availableSeats-2*j}{2} + \sum_{j=1}^k (-1)^j * (R_j * \binom{k}{j} * j! * \prod_{r=0}^{k-j-1} (\binom{availableSeats-2*j-2*r}{2}))) * 2^k$.

Можемо легко порахувати $R(j)$ використавши динамічне програмування по зломаному профілю. Вважаємо, що $m < n$, якщо ні, просто транспонуємо матрицю. Давайте пронумеруємо клітинки масиву, номер клітинки (x, y) це $(x - 1) * m + y$. $dp_{i,j,mask}$ кількість способів розставити людей у перші i клітинок, так що j пар сидять поруч так $mask$ це бінарна репрезентація останніх $m + 1$ клітинок які ми пройшли (біт 0 якщо клітинка зайнята та 1 інакше).

$$\forall j R(j) = \sum_{mask=0}^{2^{m+1}-1} dp_{n*m,j,mask}$$

Джерело: Петрозаводські збори 2020 Jagiellonian U Contest

Підготував: Максим Зуб