Задача A. Тура

Можна побачити, що відповідь — це $n + m - 2$.

Задача B. Кімната

Порахуємо, скільки плиток можна розмістити у висоту — $\lfloor \frac{n}{k} \rfloor$, у ширину — $\lfloor \frac{m}{k} \rfloor$. Тоді можемо замостити всю площу $\lfloor \frac{n}{k} \rfloor \times \lfloor \frac{m}{k} \rfloor$ плитками.

Задача C. Чесний поділ

Автор, розробник, автор розбору: Андрій Столітній

Кожна дівчинка має мати вкінці рівну кількість яблук. Нехай ця кількість буде x. Тоді

$$x + x + x = a + b + c$$

$$3x = a + b + c$$

$$x = \frac{a + b + c}{3}.$$

Тоді, якщо в дівчинки більше, ніж x яблук, вона має віддати надмір. Відповідь на задачу буде

$$\max(0, a - x), \quad \max(0, b - x), \quad \max(0, c - x).$$

Задача D. Піраміда

Автор, розробник, автор розбору: Андрій Столітній

Розглянемо піраміду по рівнях. На рівні i буде знаходитись заповнений правильний трикутник зі стороною i. На першому рівні трикутник складається з 1 кулі. Можна порахувати кількість куль на наступному рівні, знаючи кількість куль на попередньому. Додаємо до кількості куль на рівні $i - 1$ i куль — додаємо всі кулі як нову сторону. Таким чином, дві сторони подовжуються на 1, а нова сторона має довжину i.

В цій задачі обмеження дозволяли просто написати цикл і просумувати по кожному рівні. Також існує формула: $\dfrac{n(n+1)(n+2)}{6}$.

Задача E. Апельсини

Автор, розробник, автор розбору: Андрій Столітній

Розглянемо випадок, коли ми маємо зібрати всі апельсини в координаті k.

Для кожного апельсину ми маємо прийти до нього і прийти назад. Йти не в напрямку k з апельсином не є оптимальним. Тоді ми маємо для кожного апельсину повторити маршрут $k \to x_i \to k$.

Відповідь виглядає як $2\sum_{i=1}^{n}|x_i - k|$, що можна розбити на

$$2\left(\sum_{x_i \leqslant k}(k - x_i) + \sum_{x_i > k}(x_i - k)\right)$$

$$2\left(\sum_{x_i \leqslant k} k - \sum_{x_i \leqslant k} x_i + \sum_{x_i > k} x_i - \sum_{x_i > k} k\right).$$

Тоді нам треба підтримувати такі прості суми. Можемо йти від найменшого до найбільшого, тоді підтримка таких сум буде очевидно.

Загальна складність: $\mathcal{O}(n)$.

Задача F. Ділимо масив

Автор, розробник, автор розбору: Андрій Столітній

В умові задачі написано, що нам треба, щоб кожен підвідрізок довжини більше ніж 1 кожного з фінальних масивів мав невід'ємну суму. Можна помітити, що достатньо, щоб всі підвідрізки довжиною 2 та 3 мали невід'ємну суму. Тому що кожний відрізок довжини більше ніж 2 можна розбити на підвідрізки довжини 2 і 3, а якщо вони всі мають невід'ємну суму, то весь відрізок має невід'ємну суму.

Також розбиття масиву на менші масиви можна робити неявно, усуваючи елементи з масиву. Будемо підтримувати значення динамічного програмування, де $dp_{i,j}$ буде означати максимальну кількість елементів, яку ми можемо залишити з префіксу довжини i, та ми залишили останні j елементів, тобто i, $i-1, \ldots, i-j+1$. Тоді, коли додаємо новий елемент, ми можемо або залишати його, або прибирати. Так як ми не розглядаємо підвідрізки довжини більше, ніж 3, то можемо зробити стани $dp_{i,0}, dp_{i,1}, dp_{i,2}, dp_{i,3}$ —

- не взяли елемент i,
- взяли елемент i, не взяли $i-1$,
- взяли i, $i-1$, не взяли $i-2$,
- взяли i, $i-1$, $i-2$, або більше.

Тоді можна зробити такі переходи:

$$dp_{i,0} = \max\left(dp_{i-1,0},\ dp_{i-1,1},\ dp_{i-1,2},\ dp_{i-1,3}\right)$$

$$dp_{i,1} = dp_{i-1,0} + 1$$

$$dp_{i,2} = dp_{i-1,1} + 1,\ \text{якщо } a_i + a_{i-1} \geqslant 0$$

$$dp_{i,3} = \max\left(dp_{i-1,2},\ dp_{i-1,3}\right) + 1,\ \text{якщо } a_i + a_{i-1} \geqslant 0 \text{ і } a_i + a_{i-1} + a_{i-2} \geqslant 0$$

Тоді відповіддю буде

$$n - \max\left(dp_{n,0},\ dp_{n,1},\ dp_{n,2},\ dp_{n,3}\right).$$

Загальна складність: $\mathcal{O}(n)$.