Разбор задачи «Лист бумаги»

0.1 Полное решение

Всего два положения открыток, разберем каждый:

- Если располагать открытоки вертикально, то по ширине листа умещается $\lfloor \frac{x}{w} \rfloor$ открыток, а по высоте $\lfloor \frac{y}{h} \rfloor$. Значит всего открыток таким расположением можно добиться $\lfloor \frac{x}{w} \rfloor \cdot \lfloor \frac{y}{h} \rfloor$.
- Если располагать открытоки горизонтально, то по ширине листа умещается $\lfloor \frac{x}{h} \rfloor$ открыток, а по высоте $\lfloor \frac{y}{w} \rfloor$. Значит всего открыток таким расположением можно добиться $\lfloor \frac{x}{h} \rfloor \cdot \lfloor \frac{y}{w} \rfloor$.

Ответом будет являться наибольшее из двух чисел, то есть $\max(\lfloor \frac{x}{w} \rfloor \cdot \lfloor \frac{y}{h} \rfloor, \lfloor \frac{x}{h} \rfloor \cdot \lfloor \frac{y}{w} \rfloor)$.

Разбор задачи «Баскетбольный турнир»

0.2 Постановка задачи

Надо научиться разделять две ситуации.

0.3 Полное решение

- Если команды играли круговой турнир, то общее количество матчей будет равно $\frac{n \cdot (n-1)}{2}$. Соответственно, суммарное количество побед будет $\frac{n \cdot (n-1)}{2}$.
- Если команды играли олимпийскую систему, то общее количество матчей будет равно $n - 1$. Соответственно, суммарное количество побед будет $n - 1$.

Посчитаем суммарное количество побед. Если оно будет равно $\frac{n \cdot (n-1)}{2}$, то ответом будет **Round-robin**, иначе **Olympic**.

Разбор задачи «Лифт»

Данная задача связана с моделированием и реализацией функционирования лифтов.

Необходимо провести моделирование работы каждого лифта, разберем основные моменты в реализации:

1. Предположим, что лифт заполняется полностью людьми, которым необходимы этажи с номерами $a_1, \ldots, a_c$, и Тимура в лифте нет. Тогда время развозки людей представляет собой сумму времени, затраченного на остановки на каждом этаже, поездку до максимального этажа и спуск до первого этажа.

   a) Общее количество различных остановок равно количеству уникальных чисел среди $a_1, \ldots, a_c$, обозначим это как cnt_diff. Тогда время, затраченное на остановки на этажах, составляет $h \cdot cnt\_diff$.

   b) Время, затраченное на поездку до максимального этажа, равно $lift\_time \cdot (max(a_1, \ldots, a_c) - 1)$.

   c) Время, затраченное на спуск до первого этажа, также равно $lift\_time \cdot (max(a_1, \ldots, a_c) - 1)$.

   В итоге, время развозки людей равно $h \cdot cnt\_diff + 2 \cdot lift\_time \cdot (max(a_1, \ldots, a_c) - 1)$.

2. Предположим, что лифт заполняется людьми, которым нужны этажи с номерами $a_1, \ldots, a_t, n$, и в лифте присутствует Тимур (которому нужен этаж с номером n).

a) Если лифт посещает этаж n, то время на поездку до этого этажа будет равно сумме времени, затраченного на остановки на этажах до n и времени поездки до n. Количество остановок до этажа n определяется как количество уникальных чисел среди $a_1, \dots, a_t$, меньших n, обозначим это число как q. Время, затраченное на поездку до этажа n, равно $(n-1) \cdot lift\_time$. Таким образом, время развозки людей составляет $(q+1) \cdot h + (n-1) \cdot lift\_time$ (к q прибавили 1, так как Тимур выходит на n этаже).

b) Если лифт не посещает этаж n, то запустим алгоритм из пункта **a)** для этажей $n-1$ и $n+1$ соответственно. В первом случае добавим к ответу $timur\_up\_time$, во втором случае добавим к ответу $timur\_down\_time$.

Необходимо также учесть случай, когда Тимур поднимается на n этаж пешком.

Разбор задачи «Треугольники»

Для решения подзадачи, где $n \leqslant 200$, достаточно перебрать все тройки чисел $1 \leqslant a < b < c \leqslant n$, и для каждой тройки проверить, что $a + b > c$. Такие решение набирает 10 баллов.

Чтобы ускорить решение, заметим, что при фиксированных $1 \leqslant a < b \leqslant n$ нам подходят все c из отрезка $[b+1, min(n, a+b-1)]$. Значит, достаточно перебрать a, b и для каждой пары прибавить к ответу количество подходящих c, то есть $max(0, min(n, a+b-1) - b)$.

Для дальнейшей оптимизации заметим, что при фиксированном a у нас есть несколько интересных отрезков b. Первый из них соответствует решению неравенства $a+b-1 \leqslant n \Leftrightarrow b \leqslant n-a+1$. Второй — $a+b-1 > n \Leftrightarrow b > n-a+1$. Не будем забывать, что $a+1 \leqslant b \leqslant n$, то есть на самом деле отрезки выглядят так: $b \in [a+1, n-a+1]$ и $b \in [max(a+1, n-a+2), n]$. Для каждого b из первого отрезка нужно прибавить к ответу $a-1$. А для каждого b из второго отрезка значение $n-b$.

Теперь заметим, что при $a \leqslant \frac{n}{2}$ первый отрезок остается $b \in [a+1, n-a+1]$, а второй $[n-a+2, n]$. Для начала рассмотрим этот случай. Тогда для фиксированного a нужно прибавить к ответу $(n-2 \cdot a+1) \cdot (a-1) + \sum_{n-a+2 \leqslant b \leqslant n} [n-b] = (n-2 \cdot a+1) \cdot (a-1) + (a-1) \cdot n - \sum_{n-a+2 \leqslant b \leqslant n} b$. Обозначим $f(i,j)$ сумму натуральных чисел от i до j включительно. Тогда для каждого $1 \leqslant a \leqslant \frac{n}{2}$ нужно прибавить к ответу $(2 \cdot n - 2 \cdot a + 1) \cdot (a-1) - f(n-a+2, n)$.

Для случая, когда $a > \frac{n}{2}$, первый отрезок вырождается в пустой, в остальном случай разбирается аналогично.

Теперь мы получили выражение, зависящее только от a. Нетрудно видеть, что при раскрытии скобок полученное значение можно выразить через f, а также сумму $\sum_{1 \leqslant i \leqslant k} i^2 = \frac{k \cdot (k+1) \cdot (2 \cdot k+1)}{6}$. В итоге получается следующая формула $\frac{n \cdot (n+2) \cdot (2 \cdot n-5)}{24}$. Чтобы посчитать значение по модулю 10^9+7 в языкам без встроенной поддержки длинной арифметики, можно использовать расширения некоторых компиляторов, например $\_\_int128\_t$ в $g++$.