АНАЛИЗ НА РЕШЕНИЕТО НА
ЗАДАЧА CANDIES

Пълното решение на задачата може да се раздели на две части, за които се изискват коренно различни наблюдения.

I. *Първата част е свързана с комбинаторното решение на задачата.*

Тези, които са успели да напишат решението с пълно изчерпване навярно са забелязали, че при повече видове десерти отговорът на задачата е произведение на няколко числа. Нещо повече - ако отговорът за $\boldsymbol{N}$ вида десерти е бил $\boldsymbol{S}$, то при добавяне на нов вид десерт, отговорът ще е от вида $\boldsymbol{K} \times \boldsymbol{S}$, където $\boldsymbol{K}$ е естествено число. Тоест при добавяне на нов десерт, отговорът се умножава по някакво число.

За по-опитните състезатели горното наблюдение би трябвало да е достатъчно, за да изследват как се променят отговорите при различните входни данни и от тях да „изградят“ обща формула.

Но как може да изведем обща формула? За целта ще използваме една много основна идея – да си разбием задачата на по-малки и да се опитаме от тях да изведем решение за общата част. Затова нека имаме само 2 вида десерти – a от вид #1 и b от вид #2. Нека се концентрираме върху последния изяден десерт – ако беше от вид #1, то ще сме изяли последния десерт от #2 преди него $\Rightarrow$ той не може да е от #1. Тогава остава да е от #2. Забележете, че така вече както и да разпределим десертите, ще сме си осигурили изяждането на последния десерт от вид #1 преди последния от #2. Тогава останалите десерти можем да ги изядем в произволен ред, без да се опитваме да спазим някакво правило. Така задачата се преобразува в това да разпределим общо a неразличими десерта на $a + b - 1$ позиции. Или с други думи, по колко начина можем да изберем a различни от общо $a + b - 1$ позиции. Тази задача е добре известна и отговорът ѝ се дава от биномния коефициент $\binom{a+b-1}{a} = \binom{a+b-1}{b-1}$. (1)

Така успяхме да намерим формула, която щеше да работи, ако имахме само 2 вида десерти. Как обаче можем да я „разширим“ до формула, която решава нашата задача? Оказва се, че добавянето на нови десерти не променя особено идеята, по която достигнахме то формулата (1):

Нека имаме общо $\boldsymbol{N}$ вида десерти, номерирани от 1 до N, като от десерт $\boldsymbol{i}$ има A_i броя $(i = 1\,..\,N)$. Общият брой на всички десерти нека означим като Sum. Тогава последният изяден десерт трябва да е от вид $\boldsymbol{N}$. А как да разпределим останалите? Просто ще ги считаме за една голяма група и ще използваме горната формула (1). В новополучената група отново знаем, че на последното място трябва да стои десерт от вид $\boldsymbol{N-1}$. Забележете как няма значение, че вече между тези позиции ще има десерти от вид $\boldsymbol{N}$ - тях можем да ги игнорираме, те вече не влияят на нашата формула! И така, крайната формула е:

$$Ans = \binom{Sum-1}{A_N-1} \times \binom{Sum-A_N-1}{A_{N-1}-1} \times \cdots \times \binom{A_1+A_2-1}{A_2-1} \qquad (2)$$

С извеждането на тази формула си гарантираме решение, носещо 60 точки, стига да (пре)изчисляваме биномните коефициенти от формулата по разумен начин. За пълното решение на задачата ще трябва да се преборим и с:

II. *Втората част от задачата е свързана с теория на числата.*

Щом вече сме намерили формулата (2), излиза, че трябва да пресметнем умножение на до 1000 биномни коефициента. Но тъй като в последната подзадача няма допълнителни ограничения за броя на десертите от един и същ вид, то ще ни се наложи да пресмятаме биномни коефициенти от ред число до 1'000'000!

Директната формула за биномен коефициент е $\binom{N}{K} = \frac{N!}{K!\,(N-K)!}$. Нормалното приложение на тази формула обаче изглежда няма да ни е от полза, защото, дори и да съкратим $N!$ на по-голямото от $K!$ и $(N-K)!$, другото ще остане в знаменателя. И така ще търсим частното на две големи числа, като, да не забравяме, всички изчисления трябва да станат по модул 10^9+7. Наистина не звучи добре…

Оказва се, че все пак можем да използваме директната формула, съчетавайки я с няколко хитрости. Това може да стане по няколко начина:

1) Като се научим да делим по модул – и макар това в общия случай да е доста сложна задача, ако модулът P е просто число, то можем да намерим еквивалента на деление на дадено число L като намерим обратния елемент на L в този модул[1]. Този подход изисква малко повече знания, но се предполага, че най-добрите в А група са вече запознати с него и това е едно алтернативно решение за тях.

   Оттук остава само да се погрижим да не делим по веднъж за всяко число в знаменателят, а да ги събираме „накуп" – като намерим обратния елемент на $K!$ и $(N-K)!$. Така сложността на решението става $O(N \times \log 10^9 + 7)$ = $O(N)$. Реализация на това решение можете да видите в папката с материали от задачата.

2) Като разложим на прости множители числителя и знаменателя и отчетем разликата в степените за всеки множител. Реално така ще съкращаваме дробта. Тъй като биномния коефициент е винаги цяло число, то и ще можем да съкратим дробта до край.

   Нека за всяко число C поддържаме в $am[C]$ колко пъти то участва като множител в дробта. Тогава за всеки множител D от числителят ще трябва да увеличаваме с единица $am[D]$, докато за всеки множител E от знаменателят ще трябва да намаляваме с единица $am[E]$. По този начин съкратихме всички числа, които можем да съкратим директно. Проблемът е, че е възможно $am[C]$ да е отрицателно за някое число $C \Rightarrow$ трябва да разделим дробта допълнително на $C^{am[C]}$. Но ние знаем, че дробта е съкратима! Следователно трябва да продължим да разлагаме на множители числата както от числителя, така и от знаменателя до получаване на прости числа. Тогава вече всяко число от знаменателя ще си има съответно от числителя и ще можем да го премахнем (съкратим). За целта нека имаме число C. Има два случая за C:

1. C е съставно $\Rightarrow$ $C = A \times B$, $A > 1$ и $B > 1$.
   Тогава едновременното разлагане на всички срещания на C (при $am[C] < 0$ са от знаменателя, а при $am[C] > 0$ са от числителя) би станало по следния начин:
   - ➔ $am[A] \mathrel{+}= am[C]$
   - ➔ $am[B] \mathrel{+}= am[C]$
   - ➔ $am[C] = 0$
2. C е просто.
   Директно можем да умножим отговора по всички срещания на C, тоест умножаваме по $C^{am[C]}$, и да нулираме $am[C]$.

Трябва да извършваме един от двата случая, докато има число X, за което $am[\mathrm{X}] \neq 0$. Този алгоритъм за съкращаване може да се реализира най-ефективно като представим am като масив и го обходим, започвайки от най-голямата стойност на дробта и извършвайки един от двата случая за всяко поредно число докато не достигнем до 1. Други детайли по реализацията са:

- ➔ Можем в предварително изчислен масив $fact$ да пазим по един делител за всяко съставно число;
- ➔ Не трябва да въвеждаме всички срещания на всеки делител предварително, защото не би се вписало във времевия лимит. Вместо това, понеже във дробите участват само факториели, можем да поддържаме текущ брой срещания за поредното число. Ако текущото разглеждано число е C, то броят срещания за C и всички числа преди него би се увеличил с 1, ако C! участва в числителя на `(2)` и този брой би намалял с 1, ако C! участва във знаменателя на `(2)`;
- ➔ Можем да използваме бързо повдигане на степен за умножението във случай 2.

Общата сложност на това решение е $\mathrm{O}(T\ )$. Реализация на решението можете да намерите в папката с материали от задачата.

Автор: Ясен Трифонов

[1] – За повече информация за обратен елемент по модул просто число: http://en.wikipedia.org/wiki/Modular_multiplicative_inverse