АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
ТОРТИ

✓ **Тагове**: алчни стратегии, двоично търсене по отговора, малки и големи заявки

Подзадача 1 (примерен тест)

Подзадача 2

Поради малките ограничения на тази подзадача тук е възможно да се приложи експоненциално решение, което да разгледа всички възможни разпределения на N-те блата в K торти и да избере това с най-голяма височина на най-ниската измежду тях. Друг вариант е да си приложат някакви алчни стратегии, които дори и грешни, биха могли да изведат верен отговор на малките тестове. Също съществува решение със сложност $O(N^3)$, което използва част от идеите за следващите подзадачи, но поради високата си сложност, ще работи само за тестовете от тази.

Подзадача 3

Фиксираме конкретно K, за което искаме да решим задачата.

Забелязваме, че колкото повече торти се опитаме да направим, толкова по-малка ще е тяхната височина. Затова прилагаме двоично търсене по отговора (броя торти).

Нека търсим каква е максималната височина, която може да постигнем, ако искаме да направим T на брой торти. Сортираме блатовете в ненамаляващ ред на големината им и за всеки от тях търсим тортата, към която е най-оптимално да го добавим като основа. Лесно се забелязва, че това е тортата с най-малка по големина основа измежду най-ниските торти. Извършваме добавянето, ако текущият блат е достатъчно голям, за да стане основа на избраната торта, иначе не използваме този блат. За да избираме тортата с най-малка основа сред най-ниските ефективно, използваме приоритетна опашка, в която съхраняваме двойки (височина на торта; големина на основата).

Сложността на това решение е $O(N^2 * \log_2^2 N)$, единият логаритъм в сложността идва от двоичното търсене, а другият – от приоритетната опашка. Би трябвало да бъде оценено с 28 точки, но поради близките ограничения, при ефективна реализация може да хване и третата подзадача. Реализация ще намерите във файла ***cakes-n^2loglog-dobrin.cpp***.

Подзадача 4

Забелязваме, че приоритетната опашка в решението за предишната подзадача е излишно – може да поддържаме тортите в един масив и да се опитваме да добавяме нов блат за основа на тази торта, която сме разглеждали последно (тя ще е с най-малката по големина основа измежду най-ниските торти). Така получаваме решение със сложност $O(N^2 * \log_2 N)$, което получава 44 точки. То е реализирано във файла ***n2logn.cpp***.

Подзадача 5

Забелязваме, че двоичното търсене в решението за предишните две подзадачи също е излишно. Очевидно за $K = 1$, отговорът е $T = N$. Така винаги, ако знаем, че отговорът за конкретно K е T, проверяваме дали може да направим T торти с височина $K + 1$. Ако да, отговорът за $K + 1$ е T. Иначе намаляваме T с единица и пробваме отново, докато не успеем да постигнем желаната височина на тортите. Тъй като може да намалим T общо N пъти, сложността е $O(N^2)$. Реализации: ***n2.cpp*** и ***cakes_Iliyan_64p.cpp***.

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
ТОРТИ

Подзадача 6

Целта на следващата подзадача е да подскаже на състезателите, че могат да решат ефективно задачата за всички K в интервала от 100 до N.

Идеята е следната. Намираме максималната височина K_1 на една торта с алчния алгоритъм, който описахме в подзадача 3. Ясно е, че за всяко $K > K_1$ отговорът е 0. Намираме максималната височина K_2 на две торти. Отговорът за всяко $K \in [K_2 + 1, K_1]$ е равен на 1. Продължаваме по същата схема, като ако сме открили, че максималната височина на T торти е K_T, а тази на $T + 1$ торти е K_{T+1}, то отговорът за всяко $K \in [K_{T+1} + 1, K_T]$ е точно T.

Тъй като минималната височина, за която трябва да открием колко торти може да направим е 100, ни е гарантирано, че тези торти няма да бъдат повече от $T = \frac{N}{100}$. Така получаваме решение със сложност $O(\frac{N^2}{100})$.

Подзадача 7

След решаването на подзадача 6, решението на цялата задача се състои в комбинирането на двата подхода от нея и подзадача 4. Нека разделим интервала от 1 до N, за каквито стойности на K трябва да решим задачата на малки стойности, по-малки от някакво X, и големи стойности, по-големи или равни на някакво X.

Тогава ние знаем, че с решението за подзадача 4 можем да решим задачата за малките стойности на K със сложност $O(X * N * \log_2 N)$.

При големите стойности на K знаем, че отговорът няма как да е повече от $\frac{N}{X}$. Това ни дава решение със сложност $O(\frac{N^2}{X})$, както видяхме и в предишната подзадача.

Сега трябва да изберем подходяща, стойност за X, която да минимизира сумата от двете сложности. Това става, като решим уравнението:

$$X * N * \log_2 N = \frac{N^2}{X}$$

$$X * \log_2 N = \frac{N}{X}$$

$$X^2 = \frac{N}{\log_2 N}$$

$$X = \sqrt{\frac{N}{\log_2 N}}$$

Така сложността е $O\left(N * \sqrt{N * \log_2 N}\right)$. Реално стойността на X може да се различава с константен фактор от определената оптимална в зависимост от реализацията, но не би трябвало да отнеме на състезателите много опити, преди да получат решение, което се вмества в лимита за време на изпълнение.

Реализации: ***full.cpp***, ***cakes-nsqrt(nlog)-dobrin.cpp*** и ***cakes_Iliyan_100p.cpp***.

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА
ТОРТИ

Доказателство на коректността на използваната алчна стратегия:

Нека с $X \lhd Y$ да означаваме $X + C \leq Y$.

Нека сме фиксирали стойността на K и нека отговорът за това K да бъде T. Нека получените T торти да бъдат $(a_{1,1}, a_{1,2}, \ldots, a_{1,K})$, $(a_{2,1}, a_{2,2}, \ldots, a_{2,K})$, … , $(a_{T,1}, a_{T,2}, \ldots, a_{T,K})$.

Стъпка 1:

Нека да допуснем, че съществуват i и j, за които $a_{i,p} < a_{j,1}$ за $p \geq 2$. Тогава имаме тортите $(a_{i,1}, a_{i,2}, \ldots, a_{i,p}, \ldots, a_{i,K})$ и $(a_{j,1}, a_{j,2}, \ldots, a_{j,K})$, за които $a_{i,1} \lhd a_{i,2} \lhd \cdots \lhd a_{i,p} \lhd \cdots \lhd a_{i,K}$ и $a_{j,1} \lhd a_{j,2} \lhd \cdots \lhd a_{j,K}$.

Нека на тяхно място да образуваме следните две нови торти: $\big(a_{i,1}, a_{i,2}, \ldots, a_{i,p-1}, a_{j,1}, max(a_{j,2}, a_{i,p+1}), max(a_{j,3}, a_{i,p+2}), \ldots, max(a_{j,K-p+1}, a_{i,K})\big)$ и $\big(a_{i,p}, min(a_{i,p+1}, a_{j,2}), min(a_{i,p+2}, a_{j,3}), \ldots, min(a_{i,K}, a_{j,K-p+1}), a_{j,K-p+2}, \ldots, a_{j,K}\big)$.

Първата торта изпълнява условието на задачата, защото:

$$a_{i,1} \lhd \cdots \lhd a_{i,p-1} \lhd a_{i,p} < a_{j,1};$$

$$a_{j,1} \lhd a_{j,2} \leq \max(a_{j,2}, a_{i,p+1});$$

За всяко $2 \leq t \leq K - p$:

$$a_{j,t} \lhd a_{j,t+1} \leq max(a_{j,t+1}, a_{i,t+p}) \text{ и } a_{i,t+p-1} \lhd a_{i,t+p} \leq max(a_{j,t+1}, a_{i,t+p}) \Rightarrow max(a_{j,t}, a_{i,t+p-1}) \lhd max(a_{j,t+1}, a_{i,t+p}).$$

Втората торта изпълнява условието на задачата, защото:

$$a_{i,p} < a_{j,1} \lhd a_{j,2} \text{ и } a_{i,p} \lhd a_{i,p+1} \Rightarrow a_{i,p} \lhd min(a_{i,p+1}, a_{j,2});$$

За всяко $2 \leq t \leq K - p$:

$$min(a_{i,t+p-1}, a_{j,t}) \leq a_{i,t+p-1} \lhd a_{i,t+p} \text{ и } min(a_{i,t+p-1}, a_{j,t}) \leq a_{j,t} \lhd a_{j,t+1} \Rightarrow min(a_{i,t+p-1}, a_{j,t}) \lhd min(a_{i,t+p}, a_{j,t+1});$$

$$min(a_{i,K}, a_{j,K-p+1}) \leq a_{j,K-p+1} \lhd a_{j,K-p+2} \lhd \cdots \lhd a_{j,K}.$$

С помощта на описаната размяна можем да поставим всеки от най-малките T блата на най-горна позиция в някоя от тортите. Нека новите торти са $(b_{1,1}, b_{1,2}, \ldots, b_{1,K}), (b_{2,1}, b_{2,2}, \ldots, b_{2,K}), \ldots, (b_{T,1}, b_{T,2}, \ldots, b_{T,K})$, като $b_{1,1} \leq b_{2,1} \leq \cdots \leq b_{T,1}$ (ако се налага, нека ги пренаредим, за да бъдат в този ред).

АНАЛИЗ НА РЕШЕНИЕТО НА ЗАДАЧА ТОРТИ

Стъпка 2:

Нека сега да допуснем, че съществуват i и j, за които $b_{i,p} < b_{j,2}$ за $p \geq 3$. Имаме $max(b_{1,1}, b_{2,1}, \dots, b_{T,1}) < b_{i,p-1}$ (понеже най-малките T блата са на първите позиции) и $b_{i,p-1} \lhd b_{i,p}$, откъдето $b_{j,1} \lhd b_{i,p}$. Следователно можем да направим подобна размяна на тази от стъпка 1 и да получим следните две нови торти: $\left(b_{i,1}, b_{i,2}, \dots, b_{i,p-1}, b_{j,2}, max(b_{j,3}, b_{i,p+1}), max(b_{j,4}, b_{i,p+2}), \dots, max(b_{j,K-p+2}, b_{i,K})\right)$ и $(b_{j,1}, b_{i,p}, min(b_{i,p+1}, b_{j,3}), min(b_{i,p+2}, b_{j,4}), \dots, min(b_{i,K}, b_{j,K-p+2}), b_{j,K-p+3}, \dots, b_{j,K})$, които изпълняват условието на задачата.

С помощта на описаната размяна можем да поставим всеки от следващите T най-малки блата на втора позиция в някоя от тортите. Нека новите торти са $(c_{1,1}, c_{1,2}, \dots, c_{1,K}), (c_{2,1}, c_{2,2}, \dots, c_{2,K}), \dots, (c_{T,1}, c_{T,2}, \dots, c_{T,K})$, като $c_{1,1} \leq c_{2,1} \leq \cdots \leq c_{T,1}$ и $max(c_{1,2}, c_{2,2}, \dots, c_{T,2}) \leq min(c_{1,3}, \dots, c_{T,3}, c_{1,4}, \dots, c_{T,4}, \dots, c_{1,K}, \dots, c_{T,K})$.

Стъпка 3:

Нека да допуснем, че съществуват $i < j$, за които $c_{i,1} < c_{j,1} \lhd c_{j,2} < c_{i,2}$. Тогава $c_{i,1} \lhd c_{j,2}$ и $c_{j,1} \lhd c_{i,2}$ и можем да получим следните две нови торти: $(c_{i,1}, c_{j,2}, c_{j,3}, \dots, c_{j,K})$ и $(c_{j,1}, c_{i,2}, \dots, c_{i,K})$, които изпълняват условието на задачата.

С помощта на описаната размяна можем да получим тортите $(d_{1,1}, d_{1,2}, \dots, d_{1,K}), (d_{2,1}, d_{2,2}, \dots, d_{2,K}), \dots, (d_{T,1}, d_{T,2}, \dots, d_{T,K})$, като $d_{1,1} \leq d_{2,1} \leq \cdots \leq d_{T,1} \leq d_{1,2} \leq d_{2,2} \leq \cdots \leq d_{T,2}$.

Стъпка 4:

След като вече имаме, че $d_{1,1} \leq d_{2,1} \leq \cdots \leq d_{T,1} \leq d_{1,2} \leq d_{2,2} \leq \cdots \leq d_{T,2}$, можем да премахнем най-горния блат от всяка торта и да се върнем към стъпка 2, докато не ни свършат блатовете.

Заключение:

По този начин показахме, че съществува оптимално решение $(e_{1,1}, e_{1,2}, \dots, e_{1,K}), (e_{2,1}, e_{2,2}, \dots, e_{2,K}), \dots, (e_{T,1}, e_{T,2}, \dots, e_{T,K})$, в което $e_{1,1} \leq e_{2,1} \leq \cdots \leq e_{T,1} \leq e_{1,2} \leq e_{2,2} \leq \cdots \leq e_{T,2} \leq \cdots \leq e_{1,K} \leq e_{2,K} \leq \cdots \leq e_{T,K}$.

Автори: Ясен Пенчев, Добрин Башев