Зміст

Ідея розв'язку задачі A

Запропонував: Рубаненко Р.
Підготував: Міхно М.

Давайте спробуємо знайти вартість найменшого інгредієнту. Очевидно, це буде найменше число вхідного масиву, адже найменше число не можна отримати як суму деяких інших чисел. Як знайти інші елементи? Давайте припустимо, що ми встановили вартість найдешевших K елементів. Згенеруємо усі можливі суми підмножин цих K елементів та викреслимо ці суми із вхідного масиву. З елементів, що залишилися, візьмемо мінімальний. Він і буде наступним по вартості інгредієнтом.

Чому? Припустимо, на якомусь кроці ми оберемо не мінімальне число. Тоді, коли ми будемо генерувати усі суми підмножин для відповіді, ми ніколи не згенеруємо це мінімальне число, адже серед перших K чисел не буде такої підмножини (інакше мінімальне число було б викреслене на цьому кроці), а всі інші числа будуть більшими за поточний мінімум (ми візьмемо число, більше за поточний мінімум, і генеруємо послідовність-відповідь в порядку зростання).

До речі, з цього твердження випливає, що відповідь завжди встановлюється однозначно. Якщо для видалення елементів із вхідної послідовності використовувати звичайний масив, і видаляти елементи за $O($довжина масиву$)$, асимптотика буде складати $O(n \cdot 2^n \cdot 2^n)$, тому що ми робимо n ітерацій, на кожній видаляємо порядку 2^n елементів за 2^n операцій.

Якщо використовувати set, збалансоване бінарне дерево пошуку або дерево відрізків для видалення елементів, асимптотика поліпшиться до $O(n \cdot 2^n \cdot n)$, адже видалення тепер виконується за $O(log(2^n)) = n$. Але числа в задачі невеликі (до 10^6), а тому можна використовувати масив підрахунку cnt, де $cnt[x]$ зберігає кількість елементів x. Тоді видалення елементів буде виконуватись за $O(1)$, і загальна складність алгоритму стане $O(n \cdot 2^n)$, що дозволяє пройти останню групу тестів.

Асимптотика:
По часу - $O(n \cdot 2^n)$
По пам'яті - $O(\alpha)$, де α - максимальна серед вартостей наборів.

Ідея розв'язку задачі B

Запропонував і підготував: Баренблат І.

Перш за все доведемо просту лему:
Припустимо, що ми можемо обрати K міньйонів, задовольнивши при цьому умову задачі. Якщо викинути міньйона з найбільшою витривалістю, то середнє арифметичне витривалостей **не збільшиться**, а мінімальна з сил міньйонів **не зменшиться**. Таким чином, загін з $K-1$ міньйона ми також можемо утворити.

Далі у нас є кілька варіантів розв'язку:

Варіант 1

Застосуємо принцип бінарного пошуку. За лемою, якщо можна утворити загін розміру K, то і загони меншого розміру утворити теж можна, тож залишилось лише навчитись перевіряти можливість створення загону розміру K.

Для цього відсортуємо міньйонів за спаданням сили. Обираємо перших i з них (за новим порядком сортування). Мінімальну силу серед цих i міньйонів має саме i-ий. Тепер нам необхідно вибрати K мінімальних витривалостей серед i міньйонів, перевірити, чи їх середнє арифметичне не перевищує силу i. Якщо все добре - можна утворити загін з K міньйонів, інакше переходимо до $i+1$. Аби не шукати наново K мінімальних витривалостей, використаємо структуру даних, що дозволяє знаходити максимум та видаляти його. Нам підійте черга з пріоритетом, купа або ж збалансоване бінарне дерево пошуку.

Асимптотика:
По часу - $O(Nlog^2N)$
По пам'яті - $O(N)$

Варіант 2

Припустимо, що у нас є деякий набір міньйонів і ми хочемо обрати максимальну підмножину міньйонів так, щоб середнє арифметичне їх витривалостей не перевищило деякого A. Яким чином це зробити?

Уявімо собі масив, в якому i-ий елемент представляє кількість міньйонів з витривалістю рівно i. Побудуємо над цим масивом дерево відрізків, зберігаючи при цьому в його вершинах інформацію про суму витривалостей і кількість міньйонів на цьому піддереві. За нашою лемою ми можемо застосувати до цього дерева бінарний пошук, отримавши максимальну кількість міньйонів у загоні так, щоб середнє арифметичне їх витривалостей була не більша за A. Щоправда, часова складність бінарного пошуку з використанням дерева відрізків буде $O(log^2N)$, що не покращує варіант 1. Але вихід є, і називається він каскадним спуском по дереву. Детальніше можна дізнатися тут:
`http://en.wikipedia.org/wiki/Fractional_cascading` [1].

Фінальний розв'язок виглядає так: ми сортуємо міньйонів за спаданням сили, розглядаємо перших i, додаємо їх витривалості до дерева відрізків. Тепер за допомогою каскадного спуску знаходимо розмір максимального загону, середнє арифметичне витривалостей якого не перевищує силу i-го міньйона.

Щодо дерева відрізків: вхідні дані не дозволяли у явному вигляді будувати дерево відрізків, оскільки витривалості могли бути завеликими, тож передбачалося, що учасник реалізує або неявне дерево відрізків (`http://bit.ly/ImplicitSegTree`) або стиснення координат (`http://bit.ly/CoordCompression`).

Асимптотика:
Тут α - різниця між найбільшою витривалістю і найменшою витривалістю міньйонів.
По часу - $O(NlogN)$ (для варіанту з неявним деревом відрізків $O(N(logN + log\alpha))$)
По пам'яті - $O(N)$ (для варіанту з неявним деревом відрізків $O(Nlog\alpha)$).

---
[1] Тут і далі більшість статей буде англійською

Ідея розв'язку задачі C

Запропонував і підготував: Асландуков М.

Спершу помітимо, що восьминіг по своїй суті є зв'язним графом з K вершин і K ребер. Іншими словами, ми до деякого графа-дерева додали ще одне ребро, утворивши рівно один простий цикл. З цього легко знайти критерій існування восьминога: необхідно, аби сума степенів вершин була рівн $2N$, а також існувало хоча б три вершини зі степенями щонайменше 2.

Таким чином, варіант без максимізації кількості восьминогів виконується зовсім просто: ми об'єднуємо в цикл усі вершини зі степенем більшим за 2, а «листки» (зі степенем рівно 1) напряму під'єднуємо до тіла нашого восьминога (звісно, слідкуючи, куди слід прикріпляти скільки листків, щоб набарти потрібний степінь для кожної вершини).

Для вирішення задачі з максимізацією восьминогів помітимо, що завжди можна розбити восьминога з тілом довжини $L(6 \leq L)$ на двох восьминогів розмірами 3 та $L - 3$. Таким чином, можна жадібно розбити усі вершини зі степенем не меншим за 2 на групки по $3, 4$ та 5 вершин (звісно, максимізувавши їх кількість, тобто максимально групуючи по 3 вершини) і відповідним чином під'єднати вершиини зі степенем 1.

Підзадачі 1, 2

У цих підзадачах можна було помітити, що вершини степеня 3 обов'язково належать тілу восьминога, а вершини степеня 1 - «коротенькі» щупальці. Таким чином, достатньо було перевірити, що кількість вершин степеня 1 рівна кількості вершин степеня 3, а вершин степеня 3 щонайменше 3. Якщо це так, об'єднуватимемо вершини степеня 3 у цикли, а вершини степеня 1 під'єднуватимемо по одній до вершин з циклу.

Асимптотика:

По часу - $O(N)$

По пам'яті - $O(N)$

Ідея розв'язку задачі D

Запропонував: Мисак Д.
Підготували: Баренблат І., Мінаков С.

Покажемо, що задачу можна розв'язати за час $O(N\ log\ N\ +\ M)$. Домовимося про таке:

- якщо K — деяка гілка, то K' позначатиме гілку перед нею;
- якщо K — деяка гілка, а $T \leq N$, то через $T(K)$ позначимо набір із T гілок: саму гілку K та $T-1$ гілку перед нею.

Для загальності міркувань досадимо на кожну гілку без пташок по одній фіктивній пташці, що є важчою за всіх існуючих. Фіктивні пташки не зможуть впливати на рух справжніх, а самі перестрибуватимуть лише тоді, коли на їхній поточній гілці не буде жодної справжньої пташки.

Лема 1. Починаючи з N-го перестрибування (включно) N найлегших серед усіх пташок почнуть циклічно рухатися по колу.

Доведення. Без втрати загальності вважаємо, що кількість пташок дорівнює N: пташки, що не входять до числа N найлегших, не впливатимуть на їхній рух. Розглянемо тоді розташування пташок перед N-м перестрибуванням. Якби твердження леми не справджувалося, то існувала б деяка гілка K, на якій не сидить жодна пташка. Тоді перед $(N-1)$-м перестрибуванням жодна пташка не сиділа на гілці K', перед $(N-2)$-м перестрибуванням пташок не було на гілці перед K' і т. д.; перед першим перестрибуванням жодної пташки не було на гілці, наступній за К. Таким чином, на кожній фіксованій з N гілок у якийсь момент часу не було пташок. Але тоді в жоден наступний момент часу кількість пташок на цій гілці не може перевищувати 1, що означало б, що загальна кількість пташок перед N-м перестрибуванням (враховуючи щонайменше одну порожню гілку) є меншою за N. Одержали суперечність.

Наслідок 1. Якщо $T \geq\ N$, відповідь до задачі не зміниться, якщо замість M пташок початково розглядати лише $min\{N, M\}$ найлегших пташок.

Лема 2. Якщо K — деяка гілка і $T\ \leq\ N$, то пташка, що на T-му перестрибуванні стрибне з гілки К на наступну, є найлегшою серед усіх пташок, що в початковий момент часу сиділи на гілках $T(K)$ і ще не перестрибували з гілки K.

Доведення. Застосуємо метод математичної індукції. Для $T\ =\ 1$ твердження очевидне. Для $T\ >\ 1$ скористаємося твердженням леми для перестрибування $T\ -\ 1$ та гілки K'. Тоді перед T-м перестрибуванням на гілці K буде сидіти найлегша пташок з числа тих, які ще залишилися на гілках $T(K)$. Вона й перестрибне з гілки K на наступному кроці.

Лема 3. Якщо K — деяка гілка і $T\ \leq\ N$, то пташка, що на T-му перестрибуванні стрибне з гілки K на наступну, є однією з найлегших T пташок серед усіх пташок, що в початковий момент часу сиділи на гілках $T(K)$.

Доведення. Оскільки за попередні $T\ -\ 1$ перестрибування з гілки К перестрибнула не більше ніж $T\ -\ 1$ пташка з числа найлегших T, щонайменше одна з них мала залишитися. Тоді твердження випливає з леми 2. Лема 4. Якщо $T \leq N$, то множина пташок на довільній гілці K перед T-м перестрибуванням (а отже — i вага пташки, що перестрибуватиме в момент T) залежить лише від початкового розташування пташок на гілках $T(K)$. Доведення. Для $T\ =\ 1$ твердження очевидне. Для $T\ >\ 1$ двічі скористаємося твердженням леми: для моменту часу $T\ -\ 1$ та гілок K і K'. Оскільки вміст цих гілок у момент часу $T\ -\ 1$ однозначно визначає вміст гілки K у момент часу T, індукційний перехід доведено.

Наслідок 2. Якщо $T \leq N$, відповідь до задачі не зміниться, якщо замість M пташок початково розглядати лише T найлегших пташок з числа тих, що сидять на гілках $T(K)$

(якщо їх було більше ніж T). Виділити (не більше ніж) N найлегших пташок із числа (не більше ніж) M можна за лінійний від M час за допомогою ефективного пошуку порядкової статистики. Безпосередньо відповідь знайдемо тепер таким чином:

- Якщо $T \leq N$, просимулюємо твердження леми 2 за допомогою черги з пріоритетами: щораз додаватимемо до черги нову гілку з пташками та діставатимемо звідти вагу найлегшої пташки. Вийнята з черги на T-му кроці вага й буде шуканою.
- Якщо $T > N$, то підмінимо дерево з N гілок на дерево з $2N$ гілок, що містить дві «копії» початкового дерева, які циклічно переходять одна в одну (при цьому ваги птахів тепер повторюються, але це не критично, адже те, що вони різні, у розв'язку ми не використовували). Неважко зрозуміти, що і в будь-який майбутній момент часу після перельотів пташок за встановленими правилами дерево з $2N$ гілок залишатиметься ідентичною подвійною копією початкового дерева станом на той самий момент. Скористаємося цим: знайдемо відповідь для однієї з двох гілок нового дерева, що відповідають початковій гілці K, у момент часу $N + T \% N$ (де $T \% N$ позначає остачу від ділення T на N). Згідно з лемою 1 отримана відповідь буде шуканою.

Асимптотика:
По часу - $O(N\ log\ N + M)$
По пам'яті - $O(M)$