4 задача: Чай

3 модуль, 2 семестр

ФИВТ МФТИ, 2019

Описание by Илья Белов

1. Текст задачи

В одном из отделов крупной организации работает **n** человек. Как практически все сотрудники этой организации, они любят пить чай в перерывах между работой. При этом они достаточно дисциплинированы и делают в день ровно один перерыв, во время которого пьют чай. Для того, чтобы этот перерыв был максимально приятным, каждый из сотрудников этого отдела обязательно пьет чай одного из своих любимых сортов. В разные дни сотрудник может пить чай разных сортов. Для удобства пронумеруем сорта чая числами от **1** до **m**.

Недавно сотрудники отдела купили себе большой набор чайных пакетиков, который содержит $\mathbf{a_1}$ пакетиков чая сорта номер **1**, $\mathbf{a_2}$ пакетиков чая сорта номер **2**, ..., $\mathbf{a_m}$ пакетиков чая сорта номер **m**. Теперь они хотят знать, на какое максимальное число дней им может хватить купленного набора так, чтобы в каждый из дней каждому из сотрудников доставался пакетик чая одного из его любимых сортов.

Каждый сотрудник отдела пьет в день ровно одну чашку чая, которую заваривает из одного пакетика. При этом пакетики чая не завариваются повторно.

Входные данные

Первая строка содержит два целых числа **n** и **m** ($1 \le n, m \le 50$). Вторая строка содержит **m** целых чисел $\mathbf{a_1}$, ..., $\mathbf{a_m}$ ($1 \le a_i \le 10^6$ для всех **i** от **1** до **m**).

Далее следуют **n** строк — **i**-я из этих строк описывает любимые сорта **i**-го сотрудника отдела и имеет следующий формат: сначала следует положительное число $\mathbf{k_i}$ — количество любимых сортов чая этого сотрудника, а затем идут $\mathbf{k_i}$ различных чисел от **1** до **m** — номера этих сортов.

Выходные данные

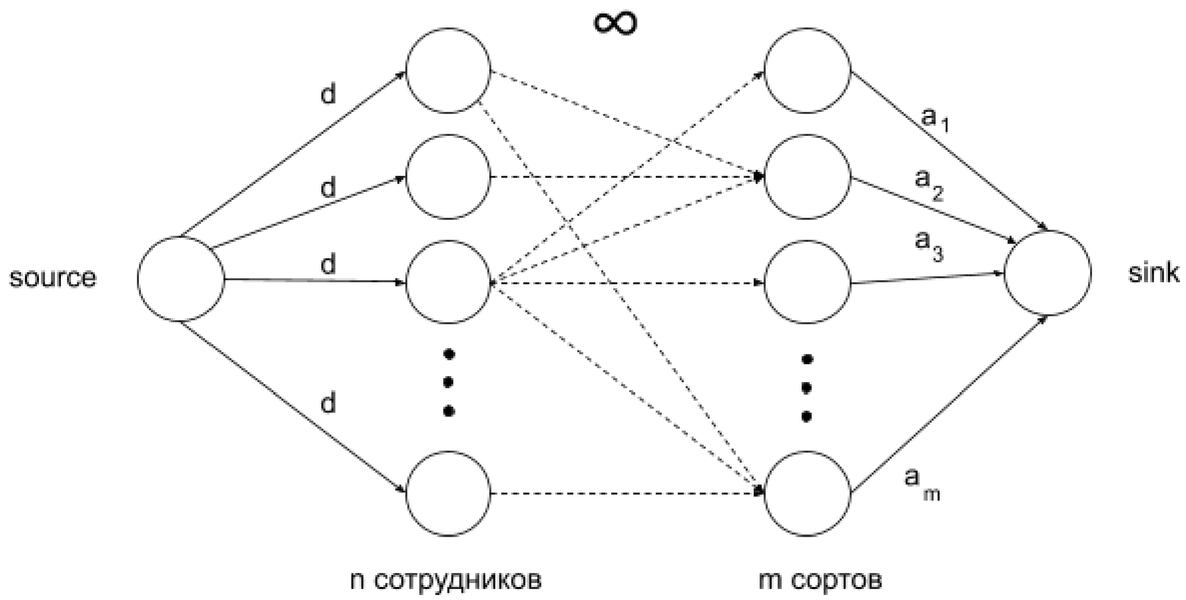
Выведите одно целое число — искомое максимальное количество дней.

Примеры:

in	out
2 3 3 2 1 2 1 2 2 1 3	3
3 3 2 7 4 2 1 2 1 2 2 2 3	4

2. Описание алгоритма

Для решения задачи построим следующую сеть:



Из источника в каждого сотрудника идёт ребро с capacity, равной кол-ву дней, в течение который сотрудники планируют испить весь чай. Из каждого сотрудника идёт по ребру capacity ∞ в любимый сорт чая. Из каждого чая в сток идёт ребро с capacity, равной кол-ву пакетиков чая такого сорта в наличии.
Будем с помощью бинпоиска перебирать d - кол-во дней, которое могут продержаться сотрудники. Максимальное количество дней, которые могут продержаться сотрудники $\sum_{i=1}^{n} \frac{a_i}{n}$ - общее кол-во пакетиков, делённое на кол-во сотрудников, минимальное - ноль.

Найдём максимальный поток через сеть. Если рёбра, выходящие из истока, оказались насыщёнными, то сотрудники пережили d-ый день и не умерли от недостатка чая в организме. Ответом на задачу будет последний такой день

3. Доказательство корректности

Докажем последнее утверждение: чая хватило на d дней *тогда и только тогда, когда* все рёбра, выходящие из истока, оказались заполнены, то есть максимальный поток через сеть оказался dn.

<u>План доказательства:</u>

$(a \leftrightarrow b) \equiv (a \to b) \wedge (b \to a) \equiv (a \to b) \wedge (\neg a \to \neg b)$

<u>Доказательство</u>:

$(a \to b)$:

Пусть у нас есть сеть, поток через которую максимален и равен dn. Пока поток через сеть не обнулился, будем искать такие пути: $s \to N \to M \to t$, где N - некоторый сотрудник и M - его некоторый любимый сорт чая, при этом поток через каждое ребро этого пути не равен нулю. Будем удалять из каждого ребра на этом пути единицу потока. Это соответствует одному акту выпивания сотрудником его любимого чая (если такой путь существует, то существует

любимый сорт M , пакетики с которым ещё не кончились). Таких путей для каждого сотрудника существует d, значит каждый сотрудник в каждый из d смог утолить жажду.
$\neg a \to \neg b$:
Пусть ребро $s \to N$ оказалось не насыщённым. Пусть поток через это ребро равен $f < d$. Опять будем уменьшать поток вдоль пути $s \to N \to M \to t$. Через f удалений не найдётся такого пути через сотрудника N, вдоль которого все рёбра имеют ненулевой поток, значит на день f не нашлось “свободного” чая для этого сотрудника. Значит, этому сотруднику не хватило чая на все d дней

4. Время работы и дополнительная память

$T = O((n+m)n^4m^2 log \sum_{i=1}^{n} \frac{a_i}{n})$ (а точнее $T = O((n+m)n^2(\sum_{i=1}^{n} k_i)^2 log \sum_{i=1}^{n} \frac{a_i}{n}))$

$M = O(nm)$

5. Доказательство времени работы и дополнительной памяти

1) Подсчитаем, как зависит количество вершин и рёбер от входных данных. Из построения сети следует: количество вершин $V = n+m+2$, количество рёбер $E = n+m+\sum_{i=1}^{n} k_i$, где k_i - количество любимых сортов чая i-того сотрудника. $\sum_{i=1}^{n} k_i = O(nm)$, поэтому в итоге получаем $E = O(nm),\ V = O(n+m)$

2) Подсчитаем, сколько времени занимает одна итерация бинпоиска. Пересчёт выходящих из истока рёбер занимает $T = nO(n)$ (одна операция ListGraph::SetCapacity стоит $O(n)$ так как из истока выходит n рёбер), то есть $T = O(n^2)$. Алгоритм Эдмондса-Карпа, использовавшийся в решении задачи, работает за $T = O(VE^2) = O((n+m)n^2m^2)$
В итоге, одна итерация бинпоиска займёт $T = O((n+m)n^4m^2)$

3) Подсчитаем итоговое время. Итераций бинпоиска произойдёт $O(log\frac{A}{n})$, где $A = \sum_{i=1}^{n} a_i$ - суммарное количество пакетиков чая всех сортов. Это число никак нельзя оценить асимптотически. В итоге получим асимптотику $T = O((n+m)n^4m^2 log \sum_{i=1}^{n} \frac{a_i}{n})$, или, если брать более точную оценку для количества рёбер, $T = O((n+m)n^2(\sum_{i=1}^{n} k_i)^2 log \sum_{i=1}^{n} \frac{a_i}{n})$

Дополнительная память потребуется для хранения остаточной сети графа. ListGraph весит $M = O(V+E) = O(n+m+nm) = O(nm)$