1 Алгоритм решения

Обозначим через V множество значений количества слитков у людей, где v_i — количество слитков у человека i, где $v_i \in [1, \bar{V}]$.

1. Инициализируем границы бинарного поиска:

$$\texttt{left\_ptr} = 0, \quad \texttt{right\_ptr} = \max\{v_i : v_i \in V\} + 1$$

2. Повторяем, пока $\texttt{right\_ptr} - \texttt{left\_ptr} > 1$:
   - Вычисляем $\texttt{mid} = \left\lfloor \frac{\texttt{left\_ptr}+\texttt{right\_ptr}}{2} \right\rfloor$;
   - Проверяем возможность перераспределения, которое определим как $I(maxFlow = \sum v_i)$ (где I — ф. индикатор), с ограничением не более чем `mid` слитков у одного человека;
   - Если возможно — обновляем `right_ptr`, иначе — `left_ptr`.
3. Ответ — минимальное допустимое значение `right_ptr`.

1.1 Проверка перераспределения через поток

Построим сеть:

- Исток s соединяется со всеми вершинами (людьми) — пропускная способность v_i;
- Из каждой вершины кроме s и t выходят два типа ребер:
  1. В другие вершины по графу доверия — с пропускной способностью $\sum v_i$ (из-за невозможности передать больше, чем в системе);
  2. В сток t — с пропускной способностью, равной `mid`.

Алгоритм поиска потока — Эдмондса-Карпа.

2 Корректность

Лемма 1. *Функция проверки возможности перераспределения монотонна по* `mid`.

Доказательство. На лекции было показано, что максимальный поток равен количеству, вытекающему из истока, и совпадает с количеством, втекающим в сток. При увеличении `mid` возрастает пропускная способность рёбер в сток, что увеличивает или сохраняет величину максимального потока. Следовательно, если перераспределение невозможно при k, то оно также невозможно при всех $k' < k$. □

Теорема 1. *В ориентированной сети, где*

- $s \to i$ *имеет ёмкость* v_i,
- $i \to t$ *имеет ёмкость* mid,
- *каждое доверительное ребро* $(u \to w)$ *— ёмкость* $\sum_{j=1}^{N} v_j$,

существует поток величины $\sum_{i=1}^{N} v_i$ *тогда и только тогда, когда можно передать все слитки между участниками так, чтобы ни у кого в любой момент не было более* mid.

Доказательство. ($\to$) Пусть есть поток f суммарно $\sum_i v_i$. Разложим его на $\sum_i v_i$ единичных путей

$$P_k : s \to i_k^{(0)} \to \cdots \to i_k^{(r_k)} \to t, \quad k = 1, \ldots, \sum_i v_i.$$

Поскольку $f(i \to t) \leqslant \mathrm{mid}$, в каждый i входит не более mid путей. Обрабатывая пути один за другим, переносим «слиток» вдоль P_k от $i_k^{(0)}$ к $i_k^{(r_k)}$. В любой момент у i не более числа путей, завершающихся в i, то есть $\leqslant \mathrm{mid}$, значит, никто не несёт свыше mid слитков.

($\leftarrow$) Пусть задана последовательность переносов, в результате у каждого i осталось d_i слитков, $d_i \leqslant \mathrm{mid}$, $\sum_i d_i = \sum_i v_i$. Определим поток:

$$f(s \to i) = v_i, \quad f(i \to t) = d_i, \quad f(u \to w) = \#\{\text{переносов } u \to w\}.$$

По доверию в сумме переносится $\leqslant \sum_j v_j$, а $d_i \leqslant \text{mid}$. В каждом i

$$v_i + \sum_u f(u \to i) = \sum_w f(i \to w) + d_i,$$

поскольку все v_i либо уходят дальше, либо остаются. Наконец, $\sum_i f(i \to t) = \sum_i d_i = \sum_i v_i$, значит, это поток нужной величины. □

Теорема 2. *Алгоритм бинарного поиска по ответу с использованием поиска потока корректен.*

Доказательство. Следует из монотонности функции проверки и корректности алгоритма Эдмондса-Карпа для нахождения максимального потока. □

3 Временная сложность

- Бинарный поиск: $O(\log \max v_i)$;
- Поиск потока Эдмондса-Карпа: $O(VE^2)$;

Общая сложность: $O(\log \max v_i \cdot VE^2)$.

4 Затраты по памяти

Хранение графа требует $O(V + E)$ памяти. Остальные расходы пренебрежимо малы.