# Международная школа ISIJ, 2019 г.

Иннополис, 2 июля 2019 г.

# Задача «Сортировка по рейтингу»

- Идея задачи — Михаил Киндер
- Решение задачи — Михаил Киндер

# Постановка задачи

- Задан граф из $n$ вершин. Каждая вершина имеет рейтинг — целое положительное число.
- Разрешается менять местами рейтинги у двух смежных вершин графа.
- Требуется отсортировать исходный набор рейтингов по возрастанию, переставляя на каждом шаге рейтинги только у двух вершин. Если это сделать невозможно, вывести -1.

**Основные темы задачи:**

- пути в графе;
- рекурсия;
- сортировка с ограничениями.

# Решение. Подзадача 1 ($n \leqslant 4$)

- Первая группа из 14 тестов.
- Несложный перебор возможных вариантов;
- Аккуратное программирование списка переставляемых пар (не более восьми шагов).

# Решение. Подзадача 3 (*boss*)

- Построим граф подчинения сотрудников (матрицу смежности);
- В подзадаче 3 есть вершина графа $B$ (boss), смежная со всеми остальными вершинами графа;
- Каждый шаг сортировки — перестановка двух сотрудников с рейтингами $x$ и $y$ за *один* или *три* шага, при этом положение остальных не изменяется.

| | |
|---|---|
| $x \ldots y \ldots B$ | $x \quad B$ |
| $B \ldots y \ldots x$ | $y \quad B$ |
| $b \ldots B \ldots x$ | $x \quad B$ |
| $y \ldots x \ldots B$ | |

# Решение. Подзадача 3 (*boss*)

**Реализация алгоритма:**

- по матрице смежности графа определяем вершину $B$, смежную со всеми остальными вершинами;
- находим вершину с *наименьшим* рейтингом $d_1$ и с помощью $\leqslant 3$ перестановок перемещаем её на первое место в массиве рейтингов $d$;
- затем находим вершину со следующим по величине рейтингом $d_2$ и ставим её на второе место указанным выше способом.
- И так далее...
- Сложность алгоритма — $O(n \log n)$. (Можно использовать и более медленные алгоритмы сортировки.)

# Решение. Подзадача 3 (*boss*)

**Реализация алгоритма:**

- по матрице смежности графа определяем вершину $B$, смежную со всеми остальными вершинами;
- находим вершину с *наименьшим* рейтингом $d_1$ и с помощью $\leqslant 3$ перестановок перемещаем её на первое место в массиве рейтингов $d$;
- затем находим вершину со следующим по величине рейтингом $d_2$ и ставим её на второе место указанным выше способом.
- И так далее...
- Сложность алгоритма — $O(n \log n)$. (Можно использовать и более медленные алгоритмы сортировки.)

# Решение. Подзадача 3 (*boss*)

**Реализация алгоритма:**

- по матрице смежности графа определяем вершину $B$, смежную со всеми остальными вершинами;
- находим вершину с *наименьшим* рейтингом $d_1$ и с помощью $\leqslant 3$ перестановок перемещаем её на первое место в массиве рейтингов $d$;
- затем находим вершину со следующим по величине рейтингом $d_2$ и ставим её на второе место указанным выше способом.
- И так далее...
- Сложность алгоритма — $O(n \log n)$. (Можно использовать и более медленные алгоритмы сортировки.)

# Решение. Подзадача 3 (*boss*)

**Реализация алгоритма:**

- по матрице смежности графа определяем вершину $B$, смежную со всеми остальными вершинами;
- находим вершину с *наименьшим* рейтингом $d_1$ и с помощью $\leqslant 3$ перестановок перемещаем её на первое место в массиве рейтингов $d$;
- затем находим вершину со следующим по величине рейтингом $d_2$ и ставим её на второе место указанным выше способом.
- И так далее...
- Сложность алгоритма — $O(n \log n)$. (Можно использовать и более медленные алгоритмы сортировки.)

# Решение. Подзадача 3 (*boss*)

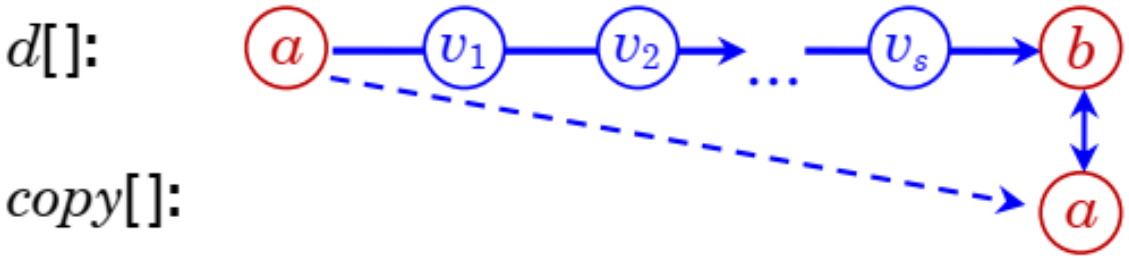
**Реализация алгоритма:**

- по матрице смежности графа определяем вершину $B$, смежную со всеми остальными вершинами;
- находим вершину с *наименьшим* рейтингом $d_1$ и с помощью $\leqslant 3$ перестановок перемещаем её на первое место в массиве рейтингов $d$;
- затем находим вершину со следующим по величине рейтингом $d_2$ и ставим её на второе место указанным выше способом.
- И так далее...
- Сложность алгоритма — $O(n \log n)$. (Можно использовать и более медленные алгоритмы сортировки.)

# Решение. Подзадача 2 ($n \leqslant 20$)

- Рассмотрим граф, вершины — это рейтинги сотрудников, а рёбра — отношения подчинённости между ними.
- Создадим вектор-копию copy[] исходного массива рейтингов и отсортируем его числа по возрастанию.
- Теперь мы знаем, на какое место в графе нужно переместить каждое число исходного набора.

Пусть $a$ — произвольная вершина графа и $b$ — вершина графа, на место которой необходимо переместить вершину $a$.



# Подзадача 2. Рекурсивное решение $O(n^2 \cdot 2^s)$

- С помощью обхода в ширину находим кратчайший путь от $a$ до $b$. Если такого пути не существует, выводим `-1`.
- Пусть этот путь состоит из $s+1$ рёбер:

$$a - v_1 - v_2 - \ldots - v_s - b.$$

- Например, если длина пути равна $1$, сотрудников $a$ и $b$ можно пересадить за один шаг; если равна $2$, — за 3 шага.
- Пусть $k$ — число шагов для перестановки $a$ и $b$, между которыми $s$ рёбер. Рассмотрим путь из $s+1$ рёбер:

$$(a - v_1 - v_2 - \ldots - v_s) - b.$$

- Выделим часть от $a$ до $v_s$ и переставим числа в вершинах $a$ и $v_s$ ($k$ шагов):

# Подзадача 2. Рекурсивное решение $O(n^2 \cdot 2^s)$

- С помощью обхода в ширину находим кратчайший путь от $a$ до $b$. Если такого пути не существует, выводим `-1`.
- Пусть этот путь состоит из $s+1$ рёбер:

$$a - v_1 - v_2 - \ldots - v_s - b.$$

- Например, если длина пути равна $1$, сотрудников $a$ и $b$ можно пересадить за один шаг; если равна $2$, — за 3 шага.
- Пусть $k$ — число шагов для перестановки $a$ и $b$, между которыми $s$ рёбер. Рассмотрим путь из $s+1$ рёбер:

$$(a - v_1 - v_2 - \ldots - v_s) - b.$$

- Выделим часть от $a$ до $v_s$ и переставим числа в вершинах $a$ и $v_s$ ($k$ шагов):

# Подзадача 2. Рекурсивное решение $O(n^2 \cdot 2^s)$

- С помощью обхода в ширину находим кратчайший путь от $a$ до $b$. Если такого пути не существует, выводим `-1`.
- Пусть этот путь состоит из $s+1$ рёбер:

$$a - v_1 - v_2 - \ldots - v_s - b.$$

- Например, если длина пути равна $1$, сотрудников $a$ и $b$ можно пересадить за один шаг; если равна $2$, — за 3 шага.
- Пусть $k$ — число шагов для перестановки $a$ и $b$, между которыми $s$ рёбер. Рассмотрим путь из $s+1$ рёбер:

$$(a - v_1 - v_2 - \ldots - v_s) - b.$$

- Выделим часть от $a$ до $v_s$ и переставим числа в вершинах $a$ и $v_s$ ($k$ шагов):

# Подзадача 2. Рекурсивное решение $O(n^2 \cdot 2^s)$

- С помощью обхода в ширину находим кратчайший путь от $a$ до $b$. Если такого пути не существует, выводим `-1`.
- Пусть этот путь состоит из $s+1$ рёбер:

$$a - v_1 - v_2 - \ldots - v_s - b.$$

- Например, если длина пути равна $1$, сотрудников $a$ и $b$ можно пересадить за один шаг; если равна $2$, — за 3 шага.
- Пусть $k$ — число шагов для перестановки $a$ и $b$, между которыми $s$ рёбер. Рассмотрим путь из $s+1$ рёбер:

$$(a - v_1 - v_2 - \ldots - v_s) - b.$$

- Выделим часть от $a$ до $v_s$ и переставим числа в вершинах $a$ и $v_s$ ($k$ шагов):

# Подзадача 2. Рекурсивное решение $O(n^2 \cdot 2^s)$

- С помощью обхода в ширину находим кратчайший путь от $a$ до $b$. Если такого пути не существует, выводим `-1`.
- Пусть этот путь состоит из $s+1$ рёбер:

$$a - v_1 - v_2 - \ldots - v_s - b.$$

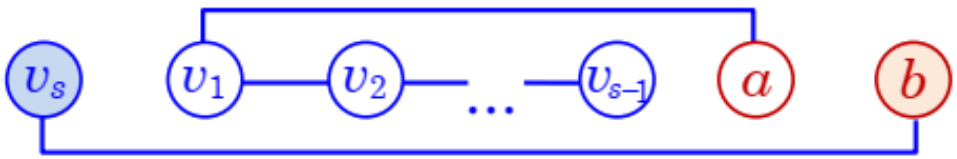
- Например, если длина пути равна $1$, сотрудников $a$ и $b$ можно пересадить за один шаг; если равна $2$, — за 3 шага.
- Пусть $k$ — число шагов для перестановки $a$ и $b$, между которыми $s$ рёбер. Рассмотрим путь из $s+1$ рёбер:

$$(a - v_1 - v_2 - \ldots - v_s) - b.$$

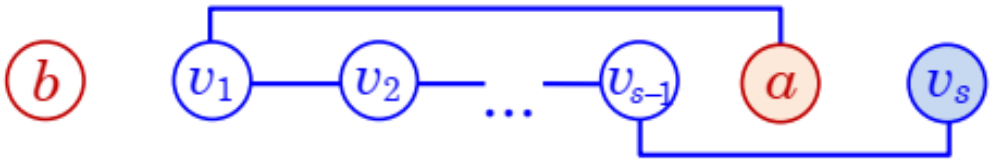
- Выделим часть от $a$ до $v_s$ и переставим числа в вершинах $a$ и $v_s$ ($k$ шагов):

# Подзадача 2. Рекурсивное решение $O(n^2 \cdot 2^s)$

- За один шаг переставим числа $b$ и $v_s$:



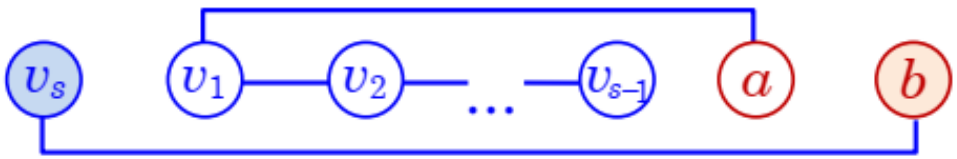

- Используем *рекурсию* и ещё за $k$ шагов поменяем числа в вершинах $a$ и $v_s$ пути $(a - v_1 - v_2 - \ldots - v_s) - b$:



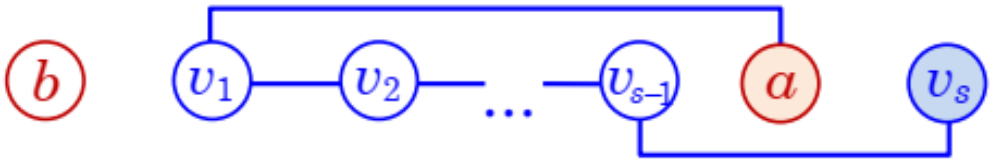

- Выделим часть от $a$ до $v_s$ и переставим числа в вершинах $a$ и $v_s$ ($k$ шагов):

# Подзадача 2. Рекурсивное решение $O(n^2 \cdot 2^s)$

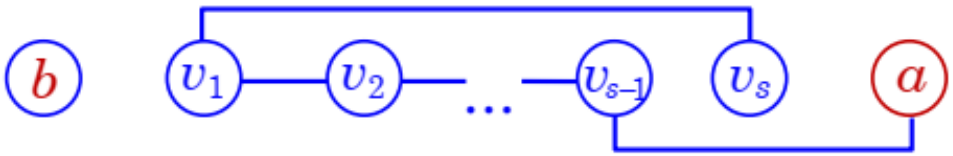
- За один шаг переставим числа $b$ и $v_s$:




- Используем *рекурсию* и ещё за $k$ шагов поменяем числа в вершинах $a$ и $v_s$ пути $(a - v_1 - v_2 - \ldots - v_s) - b$:

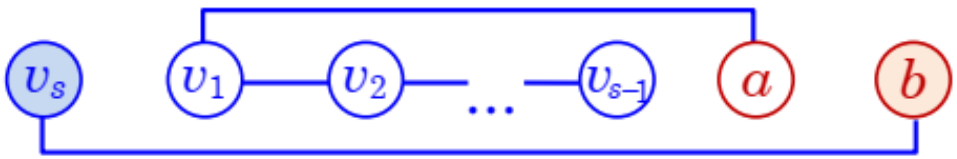



- Выделим часть от $a$ до $v_s$ и переставим числа в вершинах $a$ и $v_s$ ($k$ шагов):

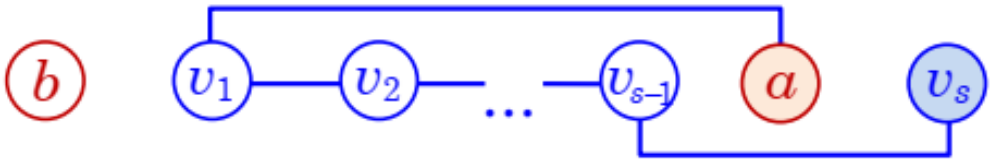


# Подзадача 2. Рекурсивное решение $O(n^2 \cdot 2^s)$

- За один шаг переставим числа $b$ и $v_s$:




- Используем *рекурсию* и ещё за $k$ шагов поменяем числа в вершинах $a$ и $v_s$ пути $(a - v_1 - v_2 - \ldots - v_s) - b$:




- Выделим часть от $a$ до $v_s$ и переставим числа в вершинах $a$ и $v_s$ ($k$ шагов):

# Подзадача 2. Рекурсивное решение $O(n^2 \cdot 2^s)$

- В этом решении требуется $k + 1 + k = 2k + 1$ шагов для перестановки двух чисел-рейтингов $a$ и $b$.
- Например, для $k = 1$ число шагов равно трём; следующее число шагов равно $2 \cdot 3 + 1 = 7$, для $k = 4$ будет $2 \cdot 7 + 1 = 15$ шагов и так далее.
- Общее число шагов равно $2^s - 1$.
- Сложность алгоритма — $O(n^2 \cdot 2^s)$, где $s$ — максимальная длина пути между вершинами $a$ и $b$ $(s < n)$.

**Замечание. Связь с задачей «Ханойские башни»**

- Задача «Ханойские башни» решается аналогичной рекурсией!
- Решение сводится к задаче с меньшими аргументами.

# Подзадача 4. Полное решение $O(n^3)$

## Циклический сдвиг массива

- Основная идея — циклический сдвиг чисел $v_i$ вдоль пути между вершинами $a$ и $b$:
- Иллюстрация на примере пути $a - v_1 - v_2 - v_3 - b$. Сначала за $s = 3$ шага выполним циклический сдвиг *вправо* первых $s + 1 = 4$ элементов этого пути:

| | |
|---|---|
| $a - v_1 - v_2 - v_3 - b$ | $a \quad v_1$ |
| $v_1 - a \quad v_2 - v_3 - b$ | $v_1 \quad v_2$ |
| $v_2 \quad a - v_1 \quad v_3 - b$ | $v_2 \quad v_3$ |
| $v_3 \quad a - v_1 - v_2 \quad b$ | |

# Подзадача 4. Полное решение $O(n^3)$

- Поменяем местами числа в вершинах $v_3$ и $b$:

$$\begin{array}{ccc|cc} v_3 & a - v_1 - v_2 & b & v_3 & b \\ \hline b & a - v_1 - v_2 - v_3 & & & \end{array}$$

- Рассмотрим путь $a - v_1 - v_2 - v_3$ в «обратном» направлении. За $s = 3$ шага выполним циклический сдвиг *влево* $s + 1 = 4$ элементов этого пути: $v_1 - v_2 - v_3 \quad a$.
- В итоге получаем нужное расположение вершин $a$ и $b$:

$$b \quad v_1 - v_2 - v_3 \quad a.$$

- Общее число шагов равно $2s - 1$, где $s$ — длина пути между вершинами $a$ и $b$.
- Итоговая сложность — $O(n^3)$.

# Решение

Вопросы?