

## Java.SE.05.Generic and Collections

### 5.1 (required)

Построить оптимальное расписание N работ на 2-х приборах по алгоритму Джонсона. В консоль вывести:

**1 обязательно:** номера работ в порядке их исполнения и время выполнения (можно вывести как в строке так и в столбик)

**2 дополнительно:** общее время производственного цикла (обработки всех деталей).

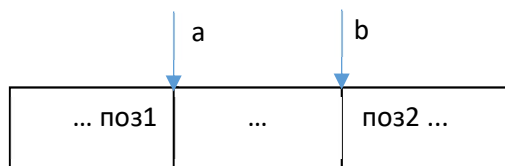
Общее время производственного цикла зависит от порядка запуска деталей в обработку. Пусть имеется N деталей, каждая должна пройти обработку сначала на первом станке, затем на втором. Для каждой детали  $D_i$  известно ее время обработки на первом и втором станке  $(t_1, t_2)_i$ . Требуется определить такой порядок запуска деталей, при котором общая длительность их обработки на обоих станках будет минимальной.

Алгоритм Джонсона:

| Номер детали                     | 1 | 2 | 3 | 4 | 5 | 5 | 6 | 7 | 8 |
|----------------------------------|---|---|---|---|---|---|---|---|---|
| t1 (время обработки на 1 станке) | 1 | 6 | 2 | 6 | 9 | 4 | 4 | 2 | 7 |
| t2 (время обработки на 2 станке) | 5 | 4 | 3 | 3 | 1 | 5 | 5 | 2 | 7 |

- Находим  $\min$  среди всех (еще не упорядоченных)  $t_1$  и  $t_2$  и соотв. номер детали  $i^*$
- Возможны варианты
  - $(t_1)_{i^*} \leq (t_2)_{i^*}$  - работу  $i^*$  ставим после позиции 1
  - $(t_1)_{i^*} > (t_2)_{i^*}$  - работу  $i^*$  ставим перед позицией 2
- Из рассмотрения исключается работа  $i^*$

Повторяем 1-3 пока все работы не будут расставлены.



Результат: 1(1,5) 3(2,3) 7(2,2) 6(4,5) 8(7,7) 2(6,4) 4(6,3) 5(9,1)

1. поскольку задание на тему Generic and collections, то максимально использовать возможности коллекций Java, в том числе:

- интерфейс Comparable<T> (или Comparator<T>)
- методы Collections.sort(); или Collections.min();
- методы ArrayList или LinkedList (для добавления/удаления элементов в список)

2. процесс перестановки можно организовать через интерфейс Queue (LinkedList<E>)

**НЕ** использовать компаратор из теории алгоритмов

### 5.2 (additional)

Дополнить задание Java.SE.01.OOP возможностями коробки и фигур:

- сортировка фигур в коробке по форме, площади и цвету. (из пленки считать цвет = цвет непокрашенных бумажных фигур)
- динамическое изменение размера коробки
- складывать фигуры в коробку и доставать из нее группами
- возможность получить из коробки:
  - все заданные фигуры (например, все треугольники)
  - все пленочные фигуры
  - все красные круги