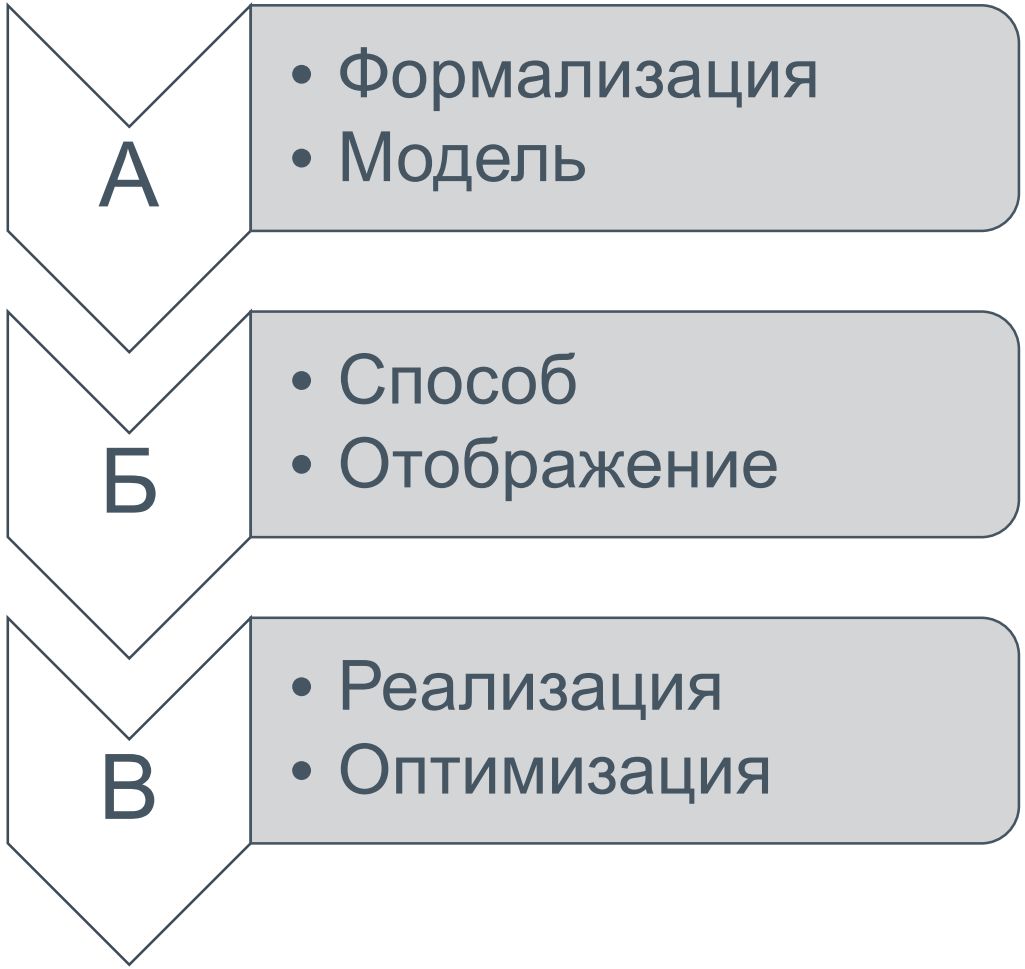
Основы алгоритмов

Лектор – профессор кафедры КММС
Дегтярев Александр Борисович

Место курса Основы алгоритмов



› Математика

› Алгоритмы

› Программирование

Изучение алгоритмов требует сочетания подходов:

- **Творческого** – для выработки идеи решения задачи
- **Логического** – для анализа правильности решения
- **Математического** – для анализа производительности
- **Скрупулёзного** – для выражения идеи в виде подробной последовательности шагов, чтобы она могла превратиться в программу

Кристофер Ван Вик, 1998

Содержание курса – 1 семестр (экзамен)

- Введение в разработку алгоритмов
- Анализ алгоритмов
- Строительные блоки алгоритмов
- Аналитические и численные алгоритмы
- Элементарные алгоритмы сортировки

Содержание курса – 2 семестр (экзамен)

- Алгоритмы сортировки
- Обход графов
- Вычислительная геометрия
- Линейная алгебра
- Интерполяция, аппроксимация и сплайны
- Алгоритмы численного дифференцирования и интегрирования
- Метод градиентного спуска

Продолжение - 2 курс «Алгоритмы и анализ данных»

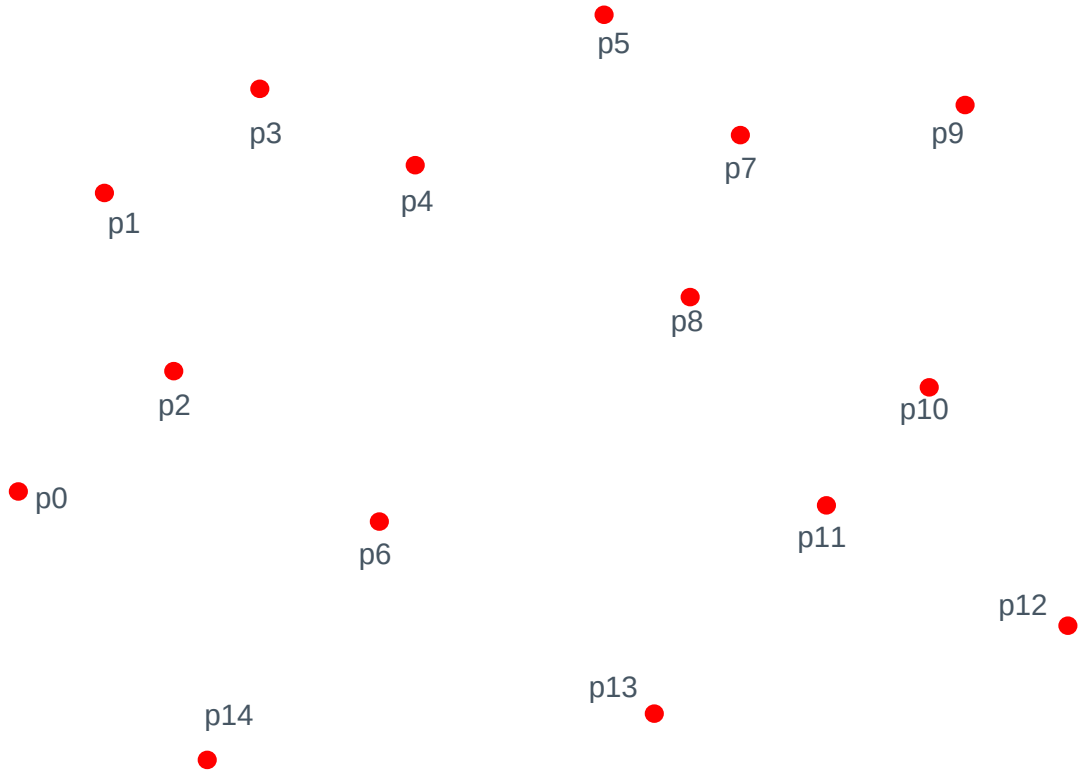
Литература

1. Скиена С.С. Алгоритмы. Руководство по разработке. СПб: БХВ-Петербург, 2018
2. Алгоритмы. Построение и анализ / Т. Х. Кормен [и др.]. – 2-е изд. – СПб.; Киев: Издательский дом "Вильямс", 2005
3. Седжвик Р. Фундаментальные алгоритмы на C++. Анализ/Структуры данных/Сортировка/Поиск – К.: Изд-во «ДиаСофт», 2001
4. Хайнеман, Д., Поллис, Г., Селков, С. Алгоритмы. Справочник с примерами на C, C++, Java и Pethon. – СПб.: ООО «Альфа-книга», 2017

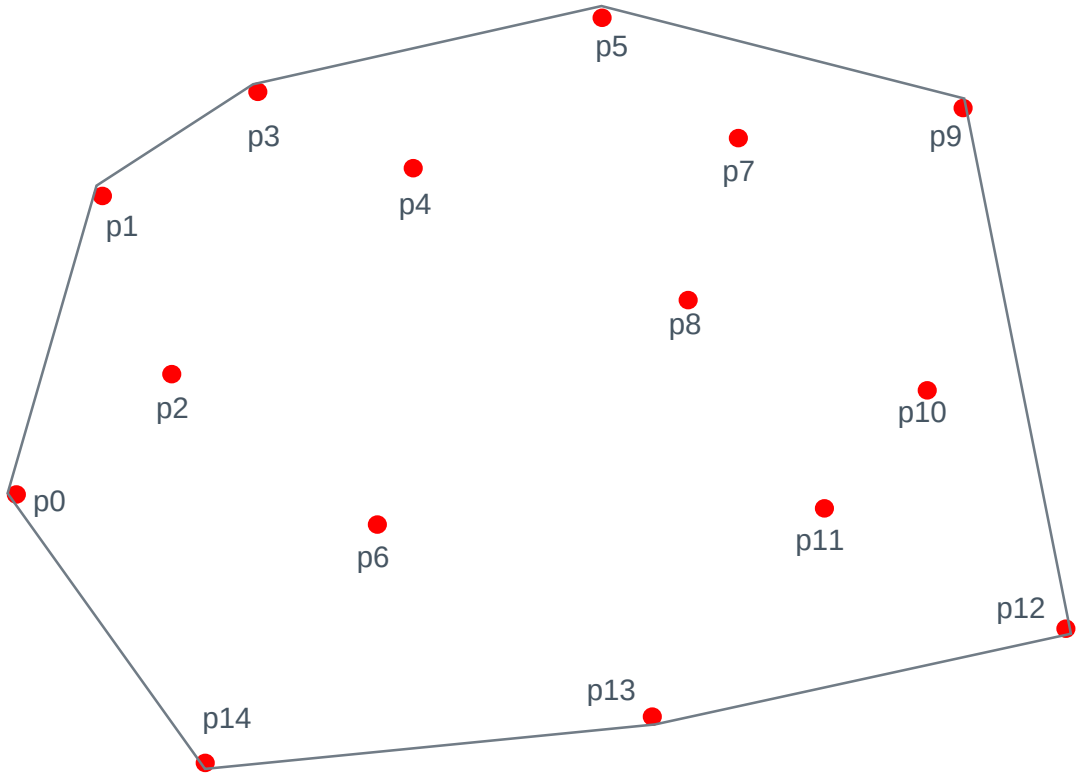
Литература (2)

5. Дж.Ортега Введение в параллельные и векторные методы решения линейных систем. – М. : Мир, 1991
6. Д. Кнут Искусство программирования для ЭВМ : в 3-х т. Том 1: Основные алгоритмы. – М. : Мир, 1976
7. Д. Кнут Искусство программирования для ЭВМ : в 3-х т. Том 3: Сортировка и поиск. – М. : Мир, 1978
8. Де Бор, К. Практическое руководство по сплайнам. – М. : Радио и связь, 1985

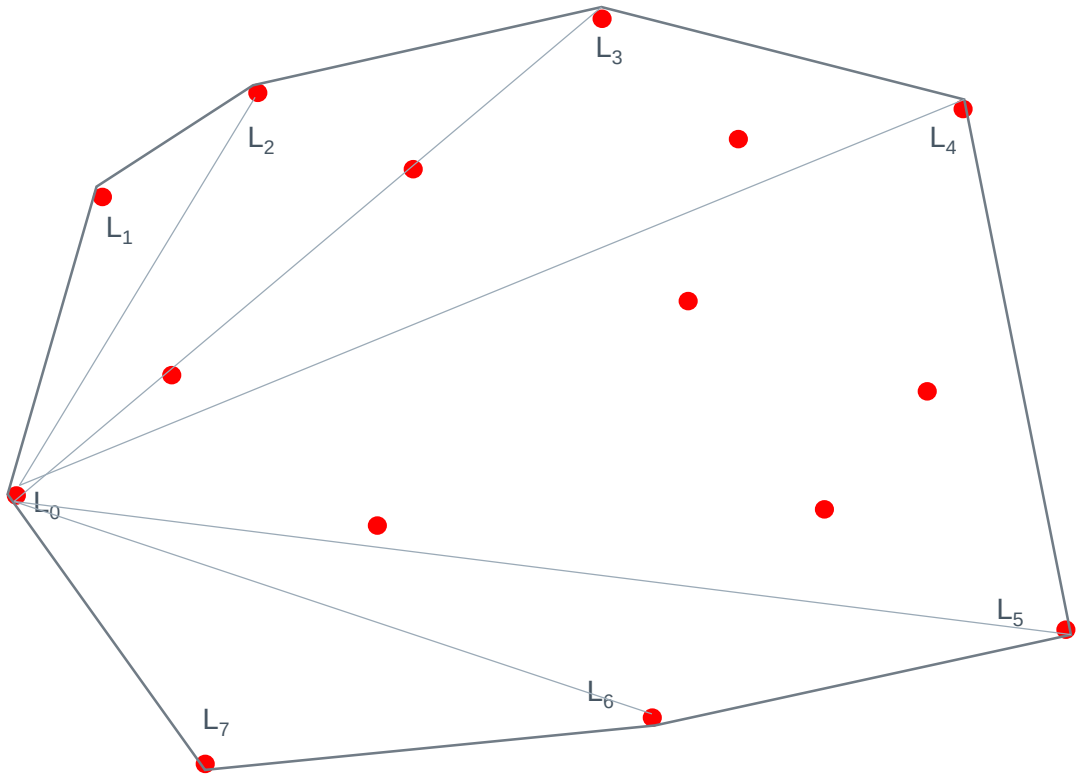
Первый пример



Первый пример



Вычисленная выпуклая оболочка



Псевдокод поиска выпуклой оболочки

Точки p0, p1, p2 образуют треугольник

```
Для каждого p0 в P
    Для каждого p1 в {P – p0}
        Для каждого p2 в {P – p0 – p1}
            Для каждого p3 в {P – p0 – p1 – p2}
                Если p3 содержится в Треугольнике(p0, p1, p2) ТО
                    Отметить p3 как внутреннюю точку
```

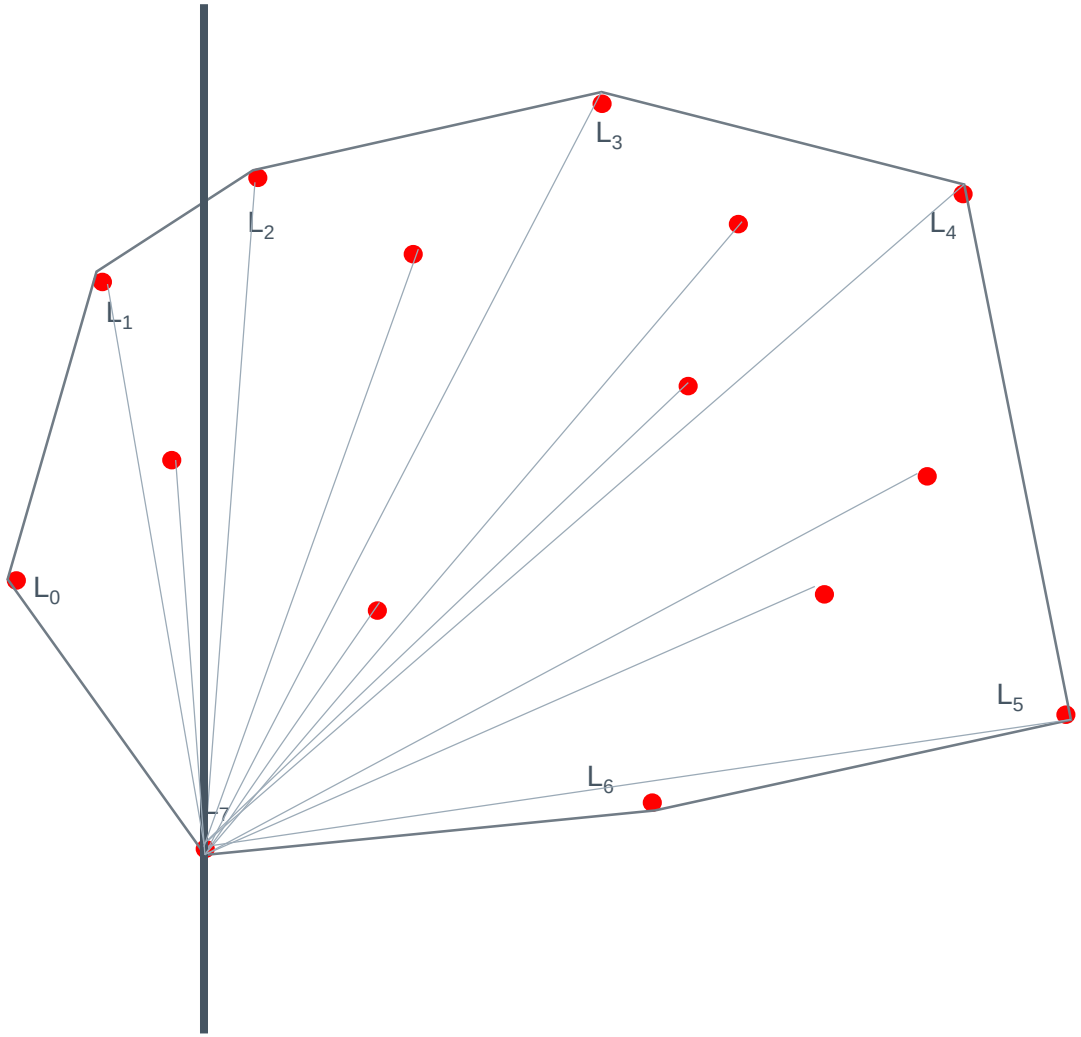
Создать массив A из всех точек P, не являющихся внутренними
Определить крайнюю слева точку множества оставшихся точек A
Отсортировать множество A по углу относительно вертикальной линии, проходящей через крайнюю слева точку

Вернуть A

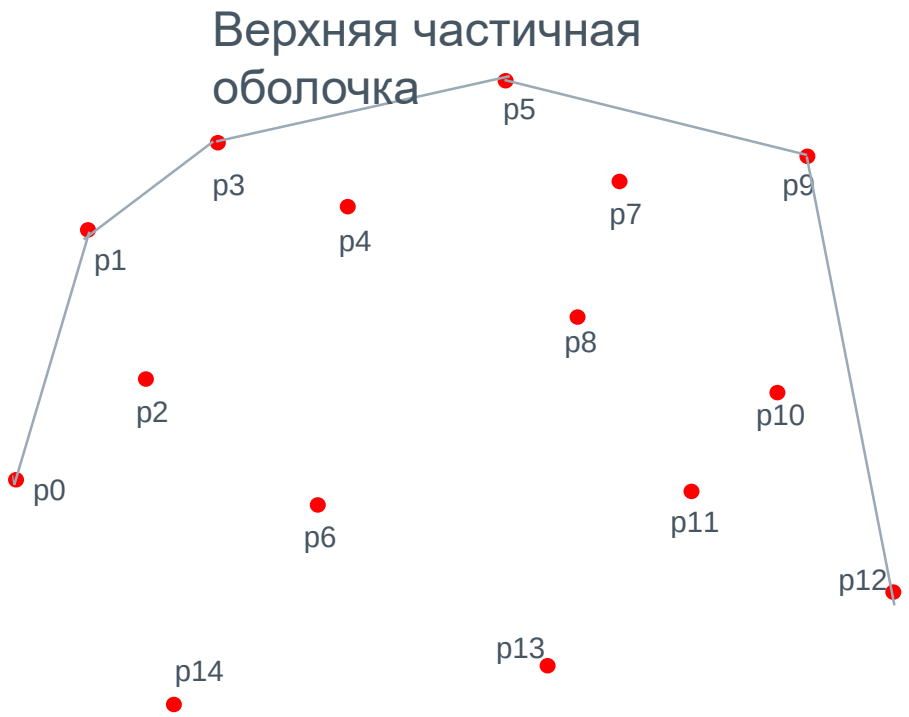
Точки, не помеченные как внутренние, образуют выпуклую оболочку

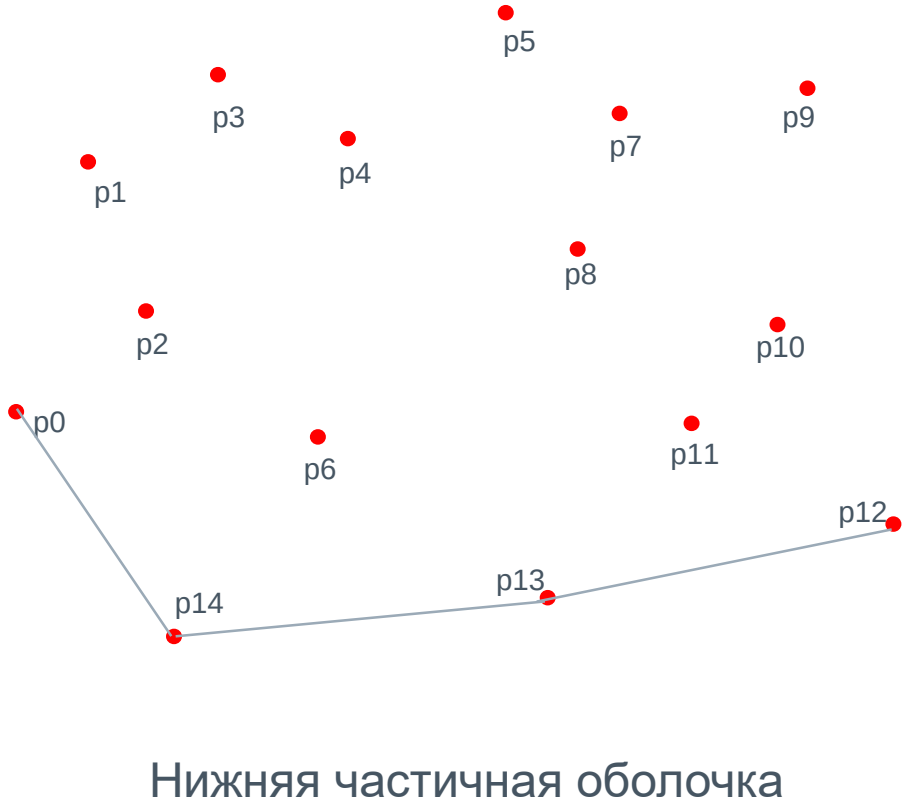
Эти углы находятся в диапазоне от -90° до 90°.

Формирование при помощи жадного алгоритма

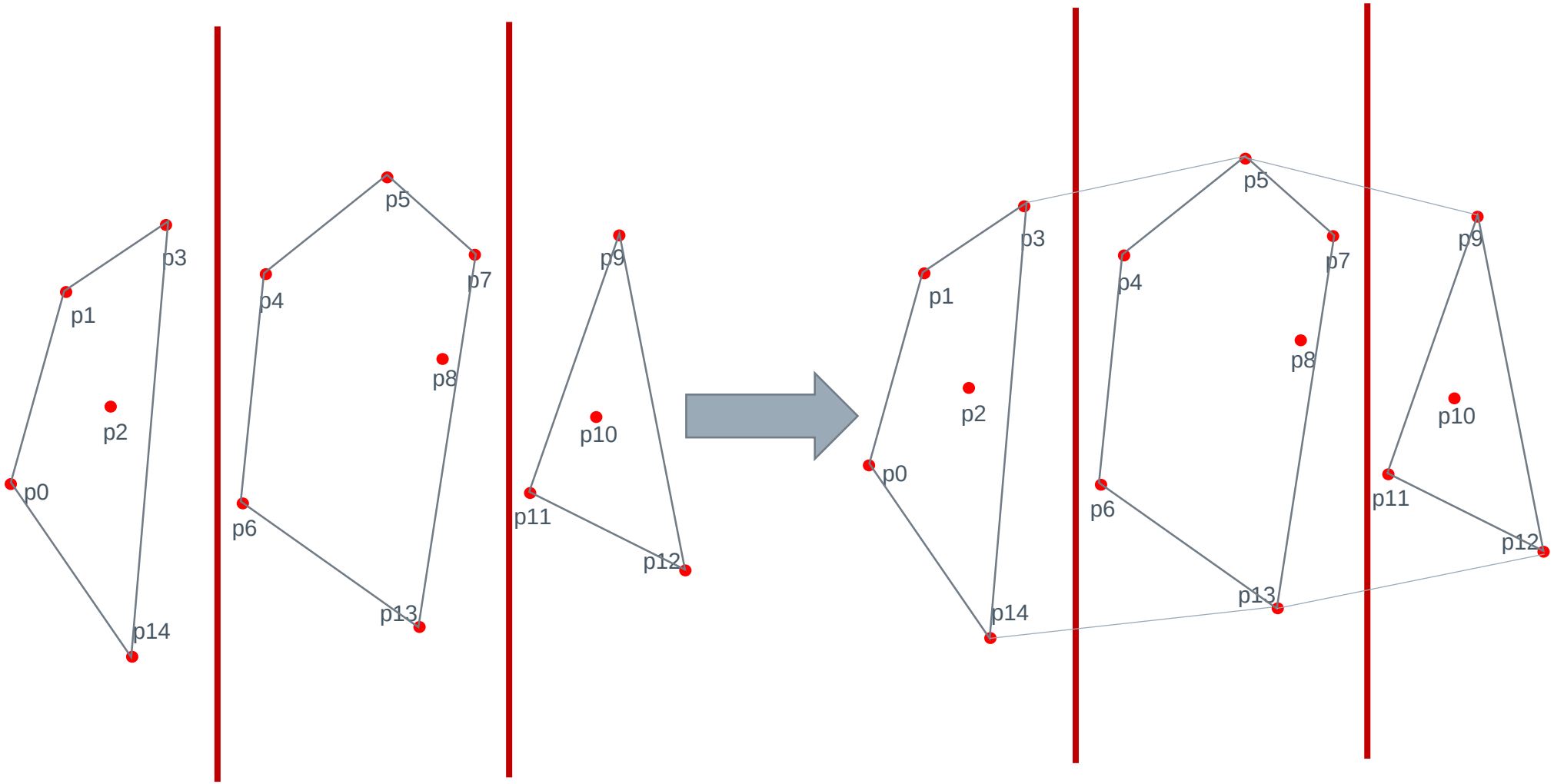


Разделяй и властвуй

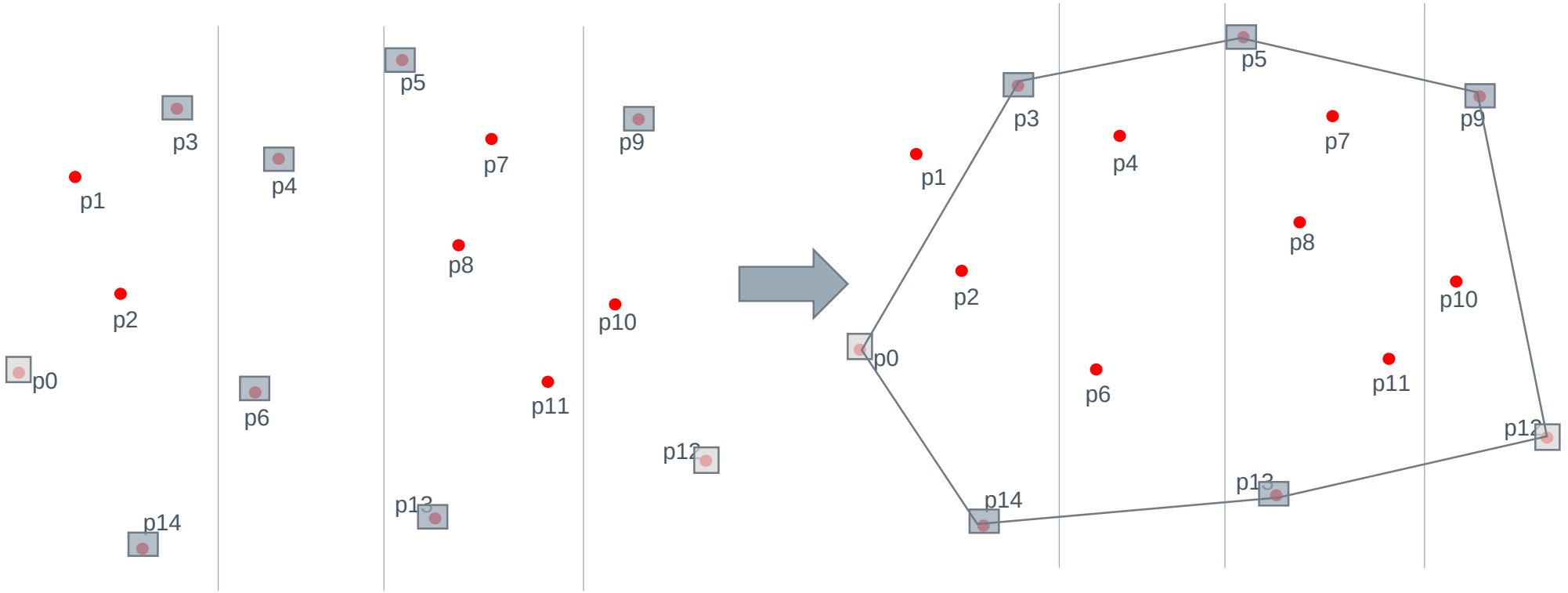




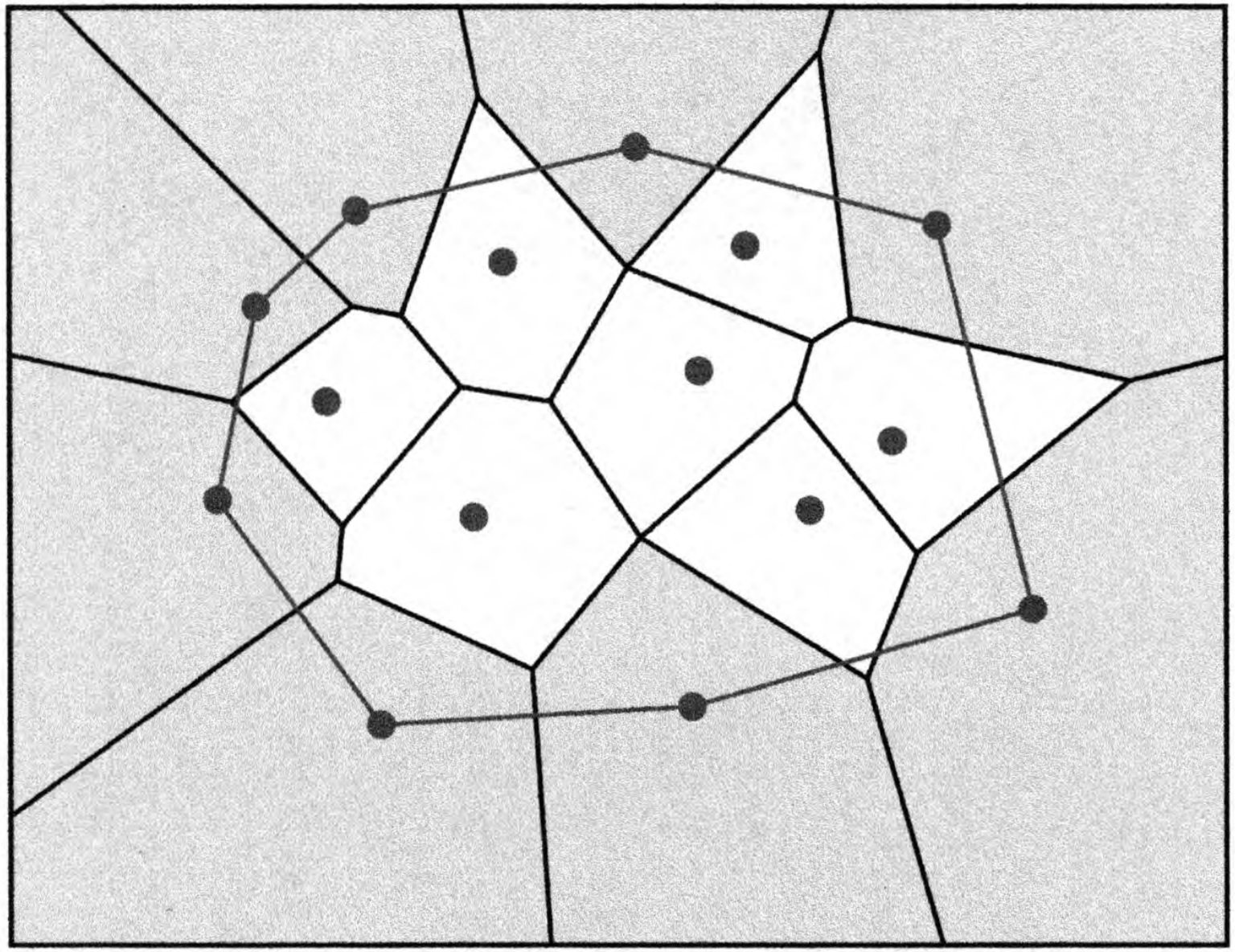
Параллельный алгоритм



Приближенный метод
Алгоритм Бентли-Фауста-Препараты



Оболочка, вычисленная из диаграммы Вороного



Что такое алгоритм?

Алгоритм (algorithm) – это формально описанная вычислительная процедура, получающая **исходные данные** (input), называемые также входом алгоритма или его аргументом, и выдающая **результат** вычислений на выход (output)

Алгоритм считается **правильным** (correct), если на любом допустимом входе он заканчивает работу и выдает результат, удовлетворяющий требованиям задачи

Представление алгоритма

Формы представления

- Обычный язык
- Псевдокод
- Язык программирования

Постановка задачи для формулировки алгоритма

Постановка задачи

› Набор допустимых входных экземпляров

› Требования к выходу алгоритма

Важный прием: сужение множества допустимых экземпляров задачи до тех пор, пока не будет найден правильный и эффективный алгоритм.

Снова о правильном и неправильном алгоритме

КОНТРПРИМЕР

Демонстрация неправильности алгоритма

Свойства хорошего *контрпримера*

› Проверяемость

› Простота

Пути поиска

› Ищите мелкомасштабные решения

› Рассмотрите все решения

› Ищите слабое звено

› Ищите ограничения

› Рассматривайте крайние случаи

Демонстрация правильности алгоритма

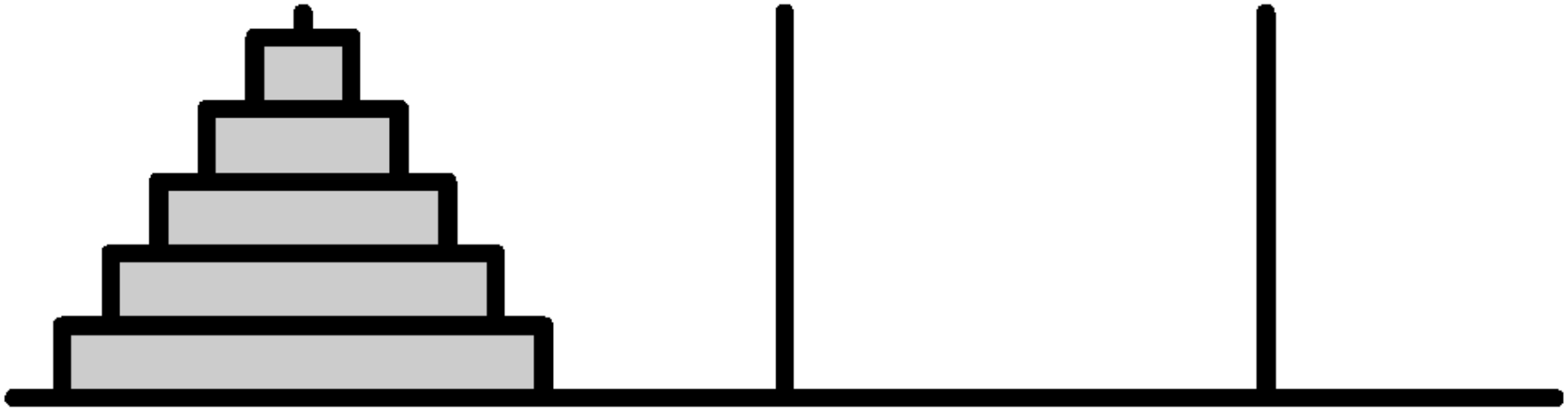
Математическая индукция

1. База индукции – Утверждение P(n) справедливо для n=1
2. Шаг индукции – Для $\forall k \in N$ из справедливости P(k) следует справедливость P(k+1)

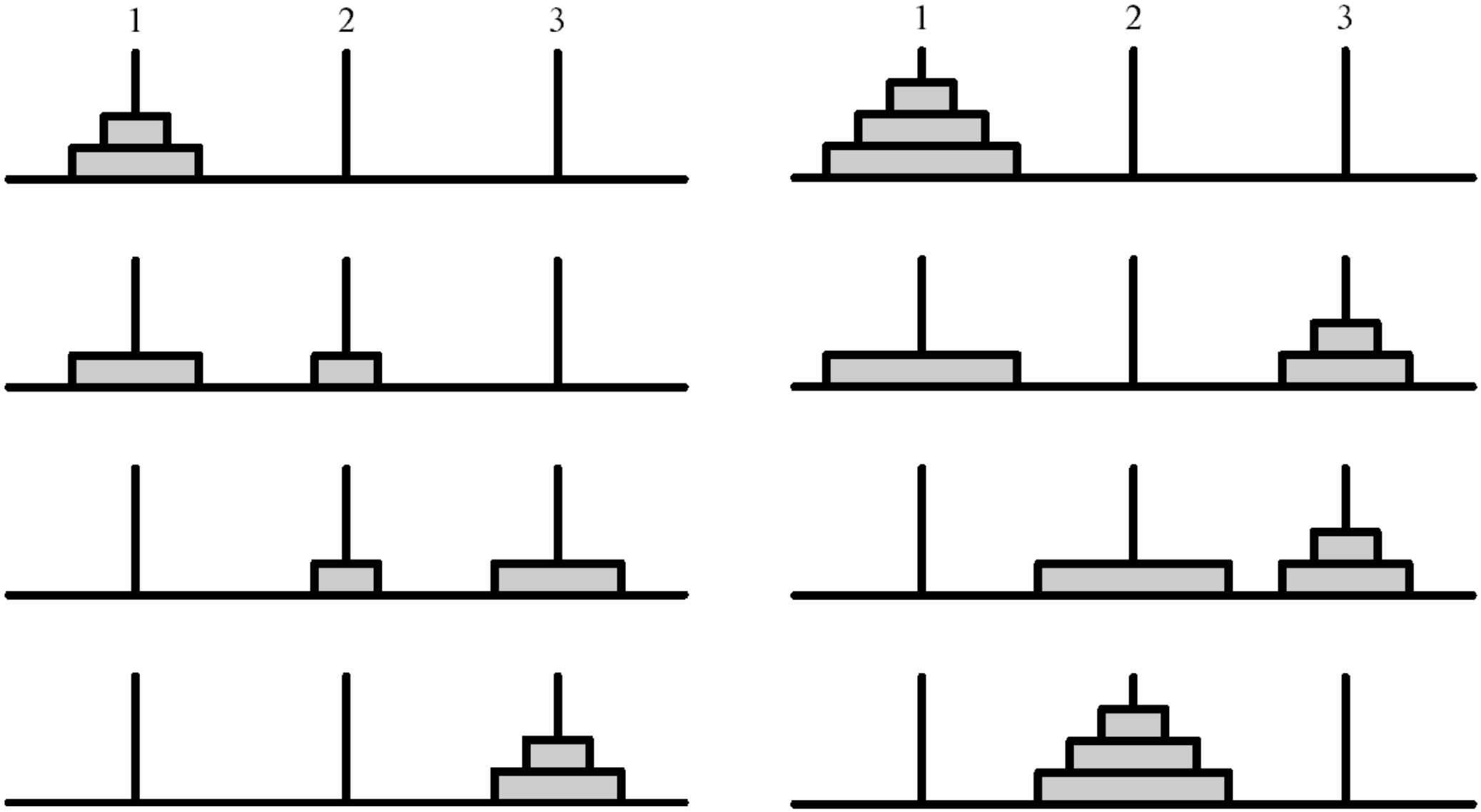
Пример. Показать, что любую сумму, начиная с 8 копеек, можно уплатить монетами 3 и 5 копеек

База		Шаг	
8 = 5+3	11 = 8+3	8+3k	8+3(k+1) = 11+3k
9 = 3+3+3	12 = 9+3	9+3k	9+3(k+1) = 12+3k
10= 5+5	13 = 10+3	10+3k	10+3(k+1)= 13+3k

Доказательство правильности алгоритма перекладки «Ханойской башни»



Доказательство правильности алгоритма перекладки «Ханойской башни» (2)



Объекты при моделировании. Комбинаторные объекты.

Стандартные структуры

- *Перестановка* – упорядоченное множество элементов
- *Подмножество* – выборка из множества элементов
- *Дерево* – иерархическое представление взаимосвязей между объектами
- *Граф* – представление взаимоотношений между произвольными парами объектов
- *Точка* – представление места в некотором геометрическом пространстве
- *Многоугольник* – представление области геометрического пространства
- *Строка* – последовательность символов или шаблонов

Объекты при моделировании. Рекурсивные объекты.

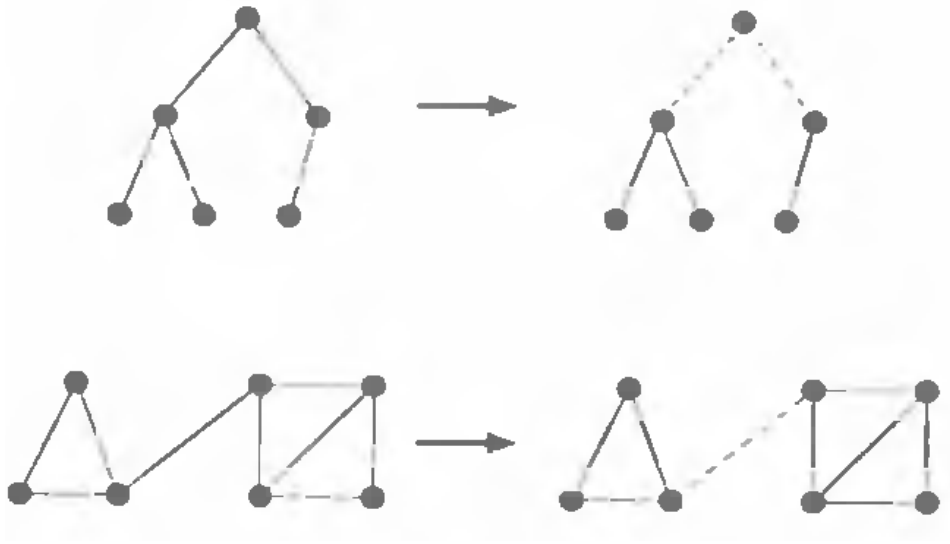
- *Перестановки* – удалив первый элемент перестановки {1,…,n}, мы получаем перестановку остающихся n-1 элементов
- *Подмножество* – каждое множество элементов {1,…,n} содержит подмножество {1,…,n-1}, являющееся результатом удаления элемента n, если такой имеется

{4, 1, 5, 2, 3} → 4 + {1, 5, 2, 3}

{1, 2, 7, 9} → 9 + {1, 2, 7}

Объекты при моделировании. Рекурсивные объекты.

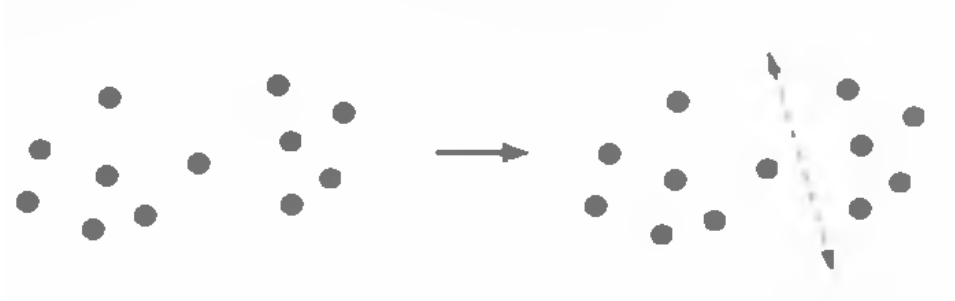
- *Деревья* – Удаление корня – коллекция меньших деревьев; Удаление какого-либо листа дерева – меньшее дерево
- *Граф* – Удаление любой вершины графа – меньший граф; Разбиение вершины графа на две группы и разрезание всех ребер, соединяющих группы – два меньших графа и набор разорванных ребер

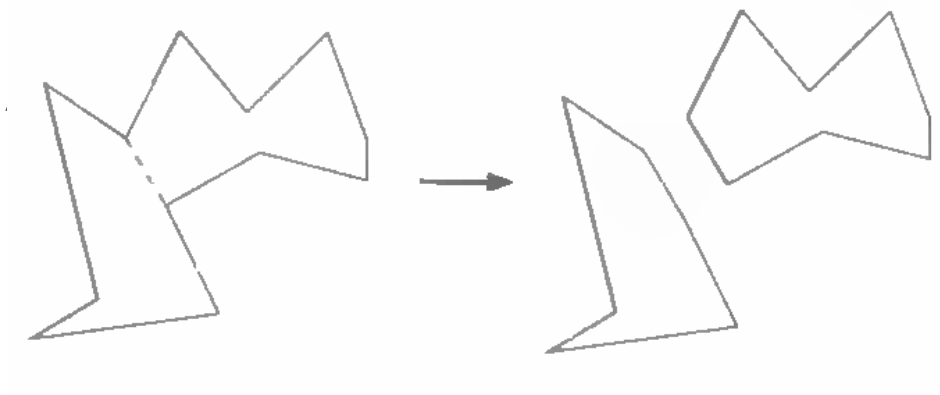


Объекты при моделировании. Рекурсивные объекты.

- *Точка* – разбиение облака точек линией – два меньших облака точек
- *Многоугольник* – соединение хордой несмежных вершин – два меньших многоугольника
- *Строка* – удаление символа в строке – меньшая строка

ДИСЦИПЛИНА → Д|ИСЦИПЛИН





Семейства алгоритмов

- *Прямое решение*
- *Жадный подход*
- *«Разделяй и властвуй»*
- *Параллельные алгоритмы*
- *Приближенный алгоритм*