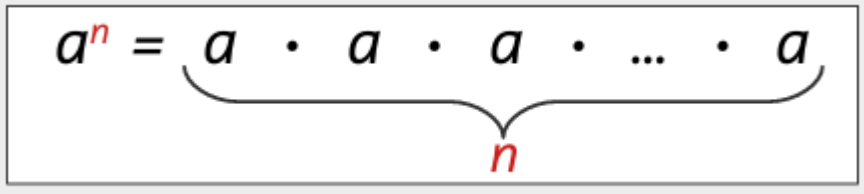
Основные методы разработки алгоритмов. Метод «грубой силы» (brute force). Метод декомпозиции. Алгоритмы, основанные на методе декомпозиции.

Основные методы разработки алгоритмов.

- **Метод грубой силы** *(brute force, исчерпывающий поиск — полный перебор)*
- **Декомпозиция** *(decomposition, «разделяй и властвуй»)*
- **Уменьшение размера задачи** («уменьшай и властвуй»)_
- **Преобразование** *(«преобразуй и властвуй»)*
- **Жадные алгоритмы** *(greedy algorithms)*
- **Динамическое программирование** *(dynamic programming)*
- **Поиск с возвратом** *(backtracking)*
- **Локальный поиск** *(local search)*

Метод "грубой силы" *(brute force)*

- **Метод грубой силы** (brute force) – решение «в лоб»
- Основан на прямом подходе к решению задачи
- Опирается на определения понятий, используемых в постановке задачи
- Пример: Задача возведения числа a в неотрицательную степень n
- Алгоритм решения «в лоб»: по определению

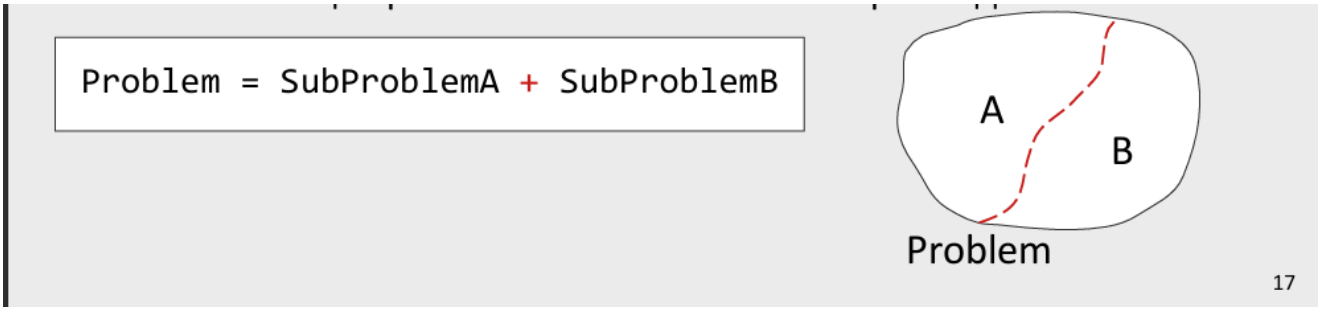


- Примеры алгоритмов, основанных на методе грубой силы:
  - Умножение матриц по определению – O(n^2)
  - Линейный поиск наибольшего/наименьшего элемента в списке
  - Сортировка выбором (selection sort) – O(n^2)

- Пузырьковая сортировка (bubble sort) – O(n^2)
- Поиск подстроки в строке методом грубой силы
- Поиск перебором пары ближайших точек на плоскости

Метод декомпозиции

- **Метод декомпозиции** *(decomposition method, «разделяй и властвуй» — «divide and conquer»)*
- Структура алгоритмов, основанных на этом методе:
  1. Задача разбивается на несколько меньших экземпляров этой же задачи
  2. Решаются сформированные меньшие экземпляры задачи *(обычно рекурсивно)*
  3. При необходимости решение исходной задачи формируется как комбинация решений меньших экземпляров задачи



- В общем случае исходная задача размера n делится на экземпляры задачи размера b, из которых a требуется решить (b > 1, a ≥ 0)
- Время T(n) работы алгоритма, основанного на методе декомпозиции, равно
  T(n) = aT(n / b) + f(n)
  где f(n) — функция, учитывающая затраты времени на разделение задачи на экземпляры и комбинирование их решений
- Рекуррентное соотношение (1) – это обобщенное **рекуррентное уравнение** декомпозиции *(general divide-and-conquer recurrence)*
- Теорема. Если в обобщенном рекуррентном уравнении декомпозиции f(n) = Θ(nd), где d ≥ 0, то вычислительная сложность алгоритма равна:

$$T(n)=\begin{cases}\Theta(n^d) & ,если\ a<b^d\\ \Theta(n^d\log n) & ,если\ a=b^d\\ \Theta(n^{\log_b a}) & ,если\ a>b^d\end{cases}$$

Примеры алгоритмов, основанных на методе декомпозиции

- Сортировка слиянием (Merge Sort)
- Быстрая сортировка (Quick Sort)
- Бинарный поиск (binary search)
- Обход бинарного дерева (tree traversal)
- Решение задачи о поиске пары ближайших точек
- Решение задачи о поиске выпуклой оболочки
- Умножение матриц алгоритмом Штрассена
