

## Задание 7. Расширенный алгоритм Евклида. Куча. Разные задачи. Графы

**1** Решите уравнения в целых числах. Нужно найти все решения, а не только частное.

1.  $238x + 385y = 133$

2.  $143x + 121y = 52$

▷

1. Найдем частное решение расширенным алгоритмом Евклида:

$$s_0 = 1, t_0 = 0$$

$$s_1 = 0, t_1 = 1$$

$$385 = 238 + 147, s_2 = 1, t_2 = -1$$

$$238 = 147 + 91, s_3 = 0 - 1 = -1, t_3 = 2$$

$$147 = 91 + 56, s_4 = 1 + 1 = 2, t_4 = -3$$

$$91 = 56 + 35, s_5 = -1 - 2 = -3, t_5 = 5$$

$$56 = 35 + 21, s_6 = 5, t_6 = -8$$

$$35 = 21 + 14, s_7 = -8, t_7 = 13$$

$$21 = 14 + 7, s_8 = 13, t_8 = -21$$

$$14 = 2 * 7$$

Получаем, что  $283 * (-21) + 385 * 13 = 7$ , откуда  $283 * (-399) + 385 * (247) = 133$ . Общее решение однородного уравнения:  $x = 55t, y = -34t, t \in \mathbb{Z}$

Общее решение имеет вид:

$$x = 55t - 283, y = -34t + 247$$

2. Найдем частное решение расширенным алгоритмом Евклида:

$$s_0 = 1, t_0 = 0$$

$$s_1 = 0, t_1 = 1$$

$$143 = 121 + 22, s_2 = 1, t_2 = -1$$

$$121 = 22 * 5 + 11, s_3 = 0 - 5 = -5, t_3 = 6$$

$$22 = 11 * 2$$

Получаем, что  $143 * (-5) + 121 * 6 = 11$ , а 52 не делится на 11. **Решений нет.**

◁

**2** Решите сравнение  $68x + 85 \equiv 0 \pmod{561}$  с помощью расширенного алгоритма Евклида. Требуется найти все решения в вычетах.

▷ Решения уравнения эквивалентно решению Диофантового уравнения  $68x + 561y = -85$

$$s_0 = 1, t_0 = 0$$

$$s_1 = 0, t_1 = 1$$

$$561 = 8 * 68 + 17, s_2 = 1, t_2 = -8$$

$$68 = 17 * 4$$

Получаем  $68 * (-8) + 561 * 1 = 17$ , откуда  $68 * (40) + 561 * (-5) = -85$

Общее решение однородного имеет вид  $x = 33t, y = -4t, t \in \mathbb{Z}$

**Ответ:**  $x \equiv 40 + 33t \pmod{561}, t = 0, \dots, 16$  ◁

**3** Найдите обратный остаток  $7^{-1} \pmod{102}$

▷ Обозначим искомый остаток за  $x$ . Тогда  $x$  является решением уравнения  $7x + 102y = 1$ .

$$s_0 = 1, t_0 = 0$$

$$s_1 = 0, t_1 = 1$$

$$102 = 14 * 7 + 4, s_2 = 1, t_2 = -14$$

$$7 = 1 * 4 + 3, s_3 = -1, t_3 = 15$$

$$4 = 3 + 1, s_4 = 2, t_4 = -29$$

Получаем  $102 * 2 + 7 * (-29) = 1$ . **Ответ:**  $7^{-1} \equiv -29 \equiv 73 \pmod{102}$  ◁

**4** Приведите алгоритм добавления элемента в уже существующую кучу на максимум из  $n$  элементов. Докажите его корректность и оцените асимптотику.

▷ Поставим новый элемент в качестве корневой вершины, т. е. его потомком будет бывшая корневая вершина. Далее шаг алгоритма следующий:

- Если оба потомка нового элемента меньше него, алгоритм останавливается.
- Иначе, меняем новый элемент с первым слева потомком, который больше его.

**Корректность:** На каждом шаге алгоритма свойство кучи может нарушаться только между новым элементом и его потомками, поэтому в момент остановки алгоритма структура действительно будет представлять из себя кучу.

**Асимптотика:** После каждого шага алгоритма позиция нового элемента понижается на один уровень. Так как алгоритм корректен, шагов не будет больше, чем уровней в куче, т. е.  $O(\log n)$ . При этом, если новый элемент меньше всех в куче, ему необходимо будет пройти все уровни, так что асимптотика:  $\Theta(\log n)$ . ◁

**5** Реализуйте очередь через два стека. Оцените асимптотику операций с получившейся очередью.

▷ ◁

**6** Сколько существует различных лесов обхода в глубину для графа-пути?

▷ За  $n$  обозначим количество вершин в графе-пути, вершины соответственно обозначим  $a_1, \dots, a_n$ . Рассмотрим произвольный лес обхода, корневые вершины его деревьев:  $a_{i_1}, \dots, a_{i_k}$ , причем последовательность индексов выбрана так, что  $i_j > i_{j+1}, \forall j = 1, \dots, k$  (здесь  $k$  — количество деревьев). Тогда дерево, начинающееся в  $a_j$  будет представлять из себя путь с вершинами  $a_j, a_j + 1, \dots, a_{j-1} - 1$ . Отдельно отметим, что  $i_k = 1$  для любого леса обхода, так как в первую вершину можно попасть только стартуя из нее. Рассмотренный лес обхода можно однозначно представить в виде двоичной последовательности длины  $n$ , в которой на местах  $i_j, j = 1, \dots$  будут стоять 1, а на остальных 0. Далее заметим, что любая такая последовательность, по написанному выше, должна начинаться с 1. Выбор остальных 1 произволен. Итак, получаем, что лесов обхода столько, сколько существует двоичных последовательностей длины  $n - 1$ .

**Ответ:**  $2^{|V|-1}$  ◁

**7** Турнир - это полный ориентированный граф, то есть такой ориентированный граф, в котором между любой парой различных вершин есть ровно одно ребро. Докажите, что в турнире на  $n$  вершинах есть простой (несамопересекающийся) путь длины  $n - 1$ . Постройте алгоритм, находящий такой путь, и оцените время его работы.

▷ Пусть умеем строить путь в турнире на  $n$  вершинах. Рассмотрим турнир  $G_{n+1}$  на  $n + 1$  вершине, в нем зафиксируем вершину  $v_{n+1}$ . В подграфе  $G_n = G_{n+1} \setminus \{v_{n+1}\}$  можем найти путь  $v_0, v_1, \dots, v_n$ . Если существует ребро  $v_{n+1}, v_0$  или  $v_n, v_{n+1}$  путь находится очевидным образом. Иначе, существуют вершины  $v_i, v_{i+1}$  такие, что существуют ребра  $v_i, v_{n+1}$  и  $v_{n+1}, v_{i+1}$ . Искомый путь:  $v_0, \dots, v_i, v_{n+1}, v_{i+1}, \dots, v_n$ .

Построим алгоритм для  $G_n$  следующим образом:

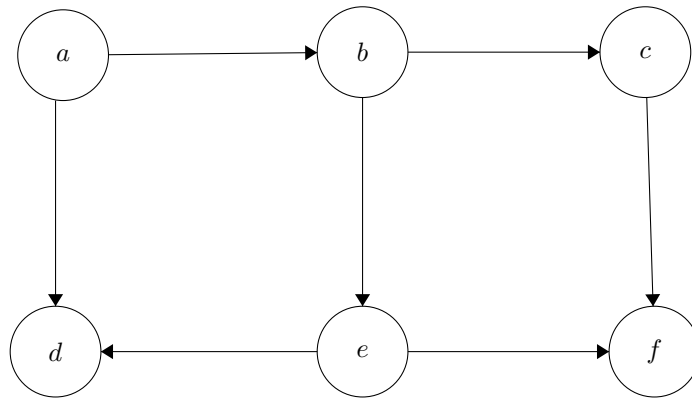
1. Выделяем произвольную вершину  $v_n$ .
2. Ищем путь в подграфе  $G_{n-1} \setminus \{v_n\}$ .
3. В этом пути ищем вершину  $v_i$  как написано выше — линейная по  $n$  операция.
4. Искомый путь как написано выше.

Обозначим за  $T(n)$  сложность алгоритма для  $G_n$ . Тогда

$$T(n) = T(n - 1) + \Theta(n)$$

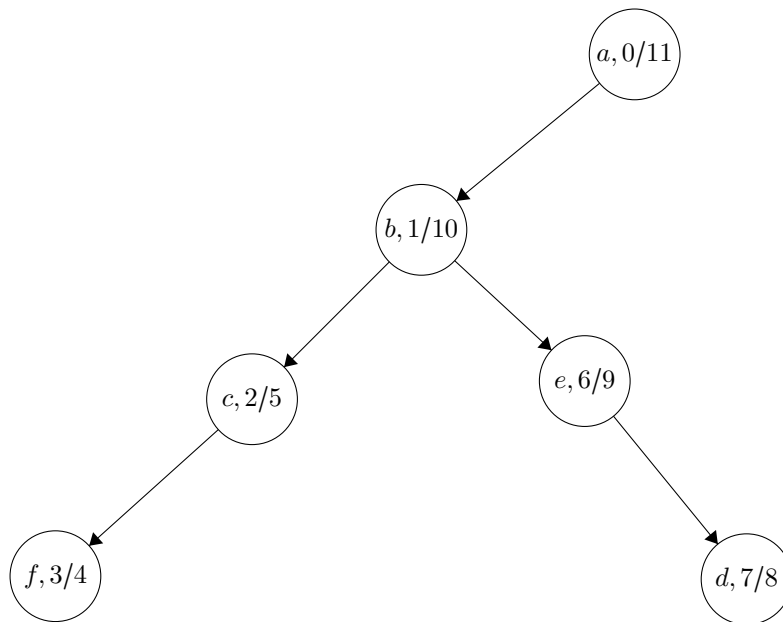
Получаем, что  $T(n) = \Theta(n^2)$ . ◁

**8** Примените DFS к графу.



Порядок выбора вершин алфавитный.

▷



◁

**9\*** На вход задачи подаются натуральные числа  $n, a_0, \dots, a_n, y$ . Необходимо определить, существует ли такое натуральное число  $x$ , что  $y = a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ .

▷ ◁

**10\*** Ваш лектор по алгоритмам нашёл два одинаковых шарика из неизвестного материала и внезапно решил измерить их прочность в этажах стоэтажного небоскрёба. Прочность равна номеру минимального этажа, при броске шарика из окна которого шарик разобьётся (максимум 100). Считаем, что если шарик уцелел, то его прочность после броска не уменьшилась. Сколько бросков шариков необходимо и достаточно для нахождения прочности?