AI Masters Algorithms **Name:** *Денис Грачев*

HomeWork 2

26 сентября 2022 г.

Задача 1

a

$238x + 385y = 133$

x	y	$238x + 385y$
0	1	385
1	0	238
-1	1	147
2	-1	91
-3	2	56
5	-3	35
-8	5	21
13	-8	14
-21	13	7
55	-34	0

Следовательно

$238a + 385b = 0 \Leftrightarrow a, b = 55k, -34k,$

$\gcd(238, 385) = 7 = -21 * 238 + 13 * 385.$

$133 : 7 = 19 \Rightarrow (-21 * 19) * 238 + (13 * 19) * 385 = 133$

$x = -21 * 19 = -399, y = 13 * 19 = 247.$

Пусть $238x' + 385y' = 133 \Rightarrow 238(x - x') + 385(y - y') = 0 \Rightarrow x', y' = x + 55k, y - 34k.$

Ответ: $x, y = -399 + 55k, 247 - 34k$, где $k \in \mathbb{Z}$.

б

$143x + 121y = 52$

x	y	$143x + 121y$
1	0	143
0	1	121
1	-1	22
-5	6	11
11	-13	0

Следовательно

$143 + 121b = 0 \Leftrightarrow a, b = 11k, -13k,$

$\gcd(143, 121) = 11 = -5 * 143 + 6 * 121.$

$52 \not\vdots 11$ Решения в целых числах не существует.

Задача 2

$68x + 85 \equiv 0 (\mod 561) \Leftrightarrow \exists N \in \mathbb{Z} : 68x + 85 = 561N \Leftrightarrow \exists N \in \mathbb{Z} : 68x + 561N = 476.$

x	N	$68x + 561N$
0	1	561
1	0	68
-8	1	17
33	-4	0

Следовательно

$68a + 561b = 0 \Leftrightarrow a, b = 33k, -4k,$

$\gcd(68, 561) = 17 = -8 * 68 + 1 * 561.$

$476 : 17 = 28 \Rightarrow (-8 * 28) * 68 + (1 * 68) * 561 = 476$

$x = (-8 * 28) = -224 \equiv 337 (\mod 561).$

Аналогично задаче1, $x = 337 + 33k = 7 + 33k$.

Ответ $x = 7 + 33k, k \in \mathbb{Z}$

Задача 3

Третий и больше столбики заполняется снизу вверх.

Я пытался выровнять знаки равенства, почему то выровнялись только mod

$$\begin{aligned}
7^{13} \mod 167 &= 7 * 7^{6^2} \mod 167 &= 7 * 81^2 \mod 167 &= 7 * 48 \mod 167 = 2 \\
7^{6} \mod 167 &= 7^{3^2} \mod 167 &= 9^2 \mod 167 &= 81 \\
7^{3} \mod 167 &= 7 * 7^2 \mod 167 &= 7 * 49 \mod 167 &= 9 \\
7^{1} \mod 167 &= 7 \mod 167 &= 7 \mod 167 &= 7
\end{aligned}$$

Задача 5

1

$T_1(n) = cn + T_1(n-1)$

$T_1(n) = cn + c(n-1) + T_1(n-2)$

$T_1(n) = cn + c(n-1) + c(n-2) + \ldots + c * 4 + T(3)$

$T_1(n) = c(n + (n-1) + (n-1) + \ldots 1) - c(3 + 2 + 1) + T(3)$

$T_1(n) = c\frac{n(n+1)}{2} - 6c + 1$

Легко видеть что $T_1(n) = \Theta(n^2)$

2

$T_2(n) = T_2(n-1) + 4T_2(n-3)$

С заменой $n = n - 3$, чтобы пропустить часть которая определяется не по формуле.

- Легко видеть, что $T_2(n)$ монотонно возрастающая функция
- $T_2(n) \geq 4T_2(n-3) \Rightarrow \frac{T_2(n)}{T_2(n-3)} \geq 4 \Rightarrow T_2(n) \geq (4^{\frac{1}{3}})^n \Rightarrow \log T_2(n) = \Omega(n)$.
- $T_2(n) \leq 5T_2(n-1) \Rightarrow \frac{T_2(n)}{T_2(n-1)} \leq 5 \Rightarrow T_2(n) \leq 5^n \Rightarrow \log T_2(n) = O(n)$

Таким образом $\log T_2(n) = \Theta(n)$

6

Пусть длина это бит в регистре n.

Сделаем функцию которая копирует k-ый бит и не меняет значения битов после k.

r - регистр, k - в соответсвии с описанием, v - значение бита записанного в k.

```
SetK(r, k, v):
    if k == 1:
        r[1] = v
        return

    SetK(r, k-1, 1)
    for i in [k-2 ... 1]:
        SetK(r, i, 0)

    r[k] = v
    return
```

Корректность

Функция рекурсивная, аргумент k с каждым вызовом уменьшается. Есть условие остановки при k=1 поэтому функция конечная.

Докажем корректность по индукции:

База

Для k=1 корректность очевидна

Переход

Пусть функция корректна для $i = 1 \dots k-1$, докажем корректность для k.

Тогда SetK(r, k-1, 1) выставит 1 в k-1 бит и не изменит биты после k.

SetK(r, i, 0) выставят 0 в i элемент и не изменит биты после i, таким образом все биты от 1 до k-1 будут 0.

Таким образом в k-1 будет стоять первая 1, и r[k] = v поставит необходимое значение в k-ый бит.

Алгоритм

```
Copy(from, to, n):
    for i in [n ... 1]:
        SetK(to, i, from[i])
```

Корректность алгоритма легко следует из корректности функции SetK.

Сложность

Обозначим сложность функции SetK(r, k, v) как $T(k)$.

Тогда $T(k) = 1 + \sum_{i=1}^{k-1} T(i)$

Докажем что $T(k) = 2^{k-1}$ по индукции.

База: для $k = 1$ верно. Шаг индукции:

Пусть верно для $i = 1 \dots k-1$, тогда

$T(k) = 1 + T(1) + T(2) + \dots T(k-1) = 1 + 1 + 2 + 4 + \dots 2^{k-2} = 2^{k-1}$

Итоговая сложность будет $T(1) + T(2) + \dots T(n) = 2^n - 1$, т.е. $\Theta(2^n)$.