Задание 2. Нижние оценки и числа Фибоначчи.

1 Найдите Θ-асимптотику функции $f(n) = \sum_{k=0}^{n} \binom{n}{k}$.

Разложим число $(1+1)^n$ по Биному Ньютона: $(1+1)^n = \sum_{i=0}^{n} \binom{n}{k} 1^{n-k} 1^k = \sum_{i=0}^{n} \binom{n}{k} \Rightarrow f(n) = 2^n$. Значит, $f(n) = \Theta(2^n)$

2 Дан набор из n монет, одна из которых фальшивая и легче других, и чашечные весы, на которые можно положить две группы монет одинакового размера и узнать, какая из них тяжелее, или что они равны по весу. Придумайте алгоритм, докажите его корректность и оцените асимптотику.

Algorithm 1 Делаем поиск монеты путем постоянно деления кучу на три равные части

```
Инициализировать текущую кучку: curr_bunch ← init_bunch
while |curr_bunch| > 1 do
    Раздели curr_bunch на три равные части A, B, C так, чтобы |A| = |B|, а |C| = ⌊|curr_bunch|/3⌋ + |curr_bunch| mod 3
    Сравниваем кучки A и B с помощью весов
    if A > B then
        curr_bunch ← B
    else if A < B then
        curr_bunch ← A
    else
        curr_bunch ← C
    end if
end while
fake_coins ← curr_bunch                ▷ в curr_bunch будет лежат только фальшивка
```

Корректность: Докажем мат.индукцией, что при таком разбиение мы будем выбирать кучку, в которой будет находится фальшивка.

$n = 0$: По условию, фальшивая монета есть в кучке, причем одна

Пусть при $n = k-1$ в выбранной кучке есть фальшивая монета. Докажем, что при $n = k$ она останется. Мы делим на три кучки на три равные части: первые две имеют одинаковое кол-во монет, оставшаяся третья такая, что может быть чуть больше(см. на алгоритм). Очевидно, что по моему предложенному алгоритму можно однозначно определить за одно взвешивание в какой из трех куч фальшивка, поэтому мы выбираем кучку в которой обязательно будет она. Так как $curr\_bunch$ каждый раз уменьшается втрое, то сходимость такого алгоритма к кучке с 1 монетой обеспечена. Значит, алгоритм корректен.

Асимптотика: т.к мы каждый раз уменьшаем размер кучки втрое, то мы сможем найти фальшивую монету за $\lceil log_3 n \rceil$, поэтому $T(n) = \Theta(log_3 n) = \Theta(log n)$

3 Найдите нижнюю оценку для задачи поиска фальшовой монеты, не предполагая ничего про алгоритм. **Нужно найти нижнюю оценку сложности именно для задачи, а не для какого-либо конкретного алгоритма, решающего ее.**

Каждый раз когда мы проводим взвешивание у нас появляется три варианта исхода: когда левая чаша тяжелее, когда правая тяжелее и когда они равны. Если рассматривать нашу задачу с точки зрения дерева решений (внутренние вершины - предикаты, а листья - ответы(какая из монет фальшивка), то для корректного нахождения фальшивой монеты дерево решений должно иметь не менее n листьев. Поскольку в троичном дереве высоты h не более 3^h листьев, необходимо $3^h \geq n$, откуда $h \geq \lceil \log_3 n \rceil$. А значит нужно как минимум $\lceil \log_3 n \rceil$ взвешиваний $\Rightarrow$ нижняя оценка сложности задачи равна $\Theta(log_3 n)$

Приведите асимптотически оптимальный алгоритм поиска фальшивой монеты. Если алгоритм из прошлой задачи уже асимптотически оптимален, ничего дополнительного делать не нужно.

————————————-УЖЕ ВЫПОЛНЕНО—————————————-

4 Найдите $7^{13} \mod 167$ с помощью быстрого возведения в степень. Нужно привести последовательность умножений и промежуточные результаты. $\mod 167$ - это остаток от деления на 167.

При вычислении $a^n \bmod m$ используем свойство модульной арифметики:

$$(a \cdot b) \bmod m = \big((a \bmod m) \cdot (b \bmod m)\big) \bmod m,$$

а также алгоритм быстрого возведения в степень слева направо, который был на семинаре.

Двоичное представление показателя: $13 = 1101_2$.

$1_{curr} \xrightarrow{1} 1^2_{curr} * 7 \xrightarrow{1} 7^2_{curr} * 7 \bmod 167 \xrightarrow{0} (7^2 * 7)^2_{curr} \bmod 167 \xrightarrow{1} [((7^2 * 7)^2)^2_{curr} \bmod 167] * 7 \bmod 167$

Рассчеты

result = 1

Бит 1: result = $(1^2 \cdot 7) \bmod 167 = 7$

Бит 1: result = $(7^2 \cdot 7) \bmod 167 = 343 \bmod 167 = 9$

Бит 0: result = $(9^2) \bmod 167 = 81$

Бит 1: result = $(81^2 \cdot 7) \bmod 167 = (6561 \bmod 167) \cdot 7 \bmod 167 = 48 \cdot 7 \bmod 167 = 336 \bmod 167 = 2$

$7^{13} \bmod 167 = 2$

Следовательно, $7^{13} \bmod 167 = 2$.

5 Рассмотрим последовательность L_i, которая похожа на числа Фибоначчи, но в обратную сторону. Она будет начинаться с двух положительных целых чисел $L_0 = a$ и $L_1 = b \leq a$, а следующие элементы будут определяться как $L_{i+1} = \max(L_i, L_{i-1}) - \min(L_i, L_{i-1})$. Определите, сколько ненулевых элементов может быть в такой последовательности, если она начинается с двух чисел, запись большего из которых состоит из n бит. Как изменится ответ, если вместо вычитания брать остаток от деления нацело?

Решение:У нас здесь представлены два алгоритма Эвклида (классический(через вычитания) и расширенный(через остатки) случаи).

Нижняя оценка для двух алгоритмов: в обоих случаях одинакова и равна 2 ненулевым числам. В качестве примера можно взять a = 5, b = 5. Тогда $(5,5) \to (5,0) \to end$. Очевидно меньше взять не получится, потому что в данной цепочке уже есть два ненулевых числа a и b.

Верхняя оценка для классического алгоритма Эвклида:

1. Идея заключается в том, что на каждом шаге такого алгоритма какое то из чисел a или b будет уменьшаться по крайней мере на 1.

2. Пусть $f(a,b)$ - - количество ненулевых членов в последовательности. Тогда после первого шага $f(a,b) = f(a-b,b) + 1 \overset{\text{п.1}}{\leq} (a-1) + 1 \leq 2^n$.

3. Пример на котором достигается верхняя оценка $a = 2^n - 1, b = a - 1 \Rightarrow (a,b) = (a, a-1) \to (1, a-1) \to \ldots \to (1,1) \to (1,0)$

Верхняя оценка для расширенного алгоритма Эвклида:

1. Идея заключается в том, что числа в последовательности остатков убывают экспоненциально. В частности, можно показать, что после каждых двух шагов алгоритма очередной остаток становится меньше, чем половина остатка два шага назад.

2. Максимальное число шагов k достигается, когда входные данные (a,b) являются парой последовательных чисел Фибоначчи, (F_m, F_{m-1}), так как в этом случае все частные от деления равны 1, что обеспечивает максимально медленное убывание. Для такой пары последовательность ненулевых членов будет $F_m, F_{m-1}, \ldots, F_2, F_1$. Таким образом, чтобы найти максимальное число членов, нужно найти максимальный возможный индекс m.

3. Максимальный индекс m_{max} определяется условием, что $F_{m_{max}}$ должно быть представимо n битами, то есть $F_{m_{max}} < 2^n$. Используя формулу Бине $F_m \approx \frac{\phi^m}{\sqrt{5}}$, где $\phi \approx 1.618$, получаем:

$$\frac{\phi^{m_{max}}}{\sqrt{5}} < 2^n \Rightarrow m_{max} < \frac{n \ln 2 + \ln \sqrt{5}}{\ln \phi} \approx 1.44n + 1.16$$

Поскольку количество членов в последовательности для пары (F_m, F_{m-1}) равно $m-1$ (от F_m до F_1), то максимальное число ненулевых членов равно $m_{max} - 1 = \lfloor \frac{n \ln 2 + \ln \sqrt{5}}{\ln \phi} \rfloor - 1$.