A3

```cpp
int x = 100;
int y = 0;

for (size_t outer = 1; outer <= n; outer *= 2) {
    x = x + outer;
    for (size_t inner = 2; inner < n; ++inner) {
        if (x > y / inner) y = y + outer / inner;
        else --y;
    }
}
```

1. Составьте точное выражение для функции временной сложности T(n) с учетом того, что арифметическая операция, присваивание и сравнение считаются одной элементарной операцией (каждая). В ответе представьте ход вычислений.

внутри условного оператора 2 операции: деление и сравнение

при выполнении сравнения нужно выполнить 3 операции

иначе 1 операция

давайте докажем что если во внутреннем цикле условие выполнилось хотя бы 1 раз, то оно будет выполнятся все следующие разы

и так у нас один раз выполнилось что $x > \frac{y}{inner}$

тогда сравним

$lim_{n\to\infty}\frac{y+x\sum_k^n\frac{1}{i}}{n} = \lim\frac{y}{n} + x\frac{\sum\frac{1}{i}}{n}$ и x

перенесем сумму в правую часть и получим

$\frac{y}{n}$ и $x(1-\frac{\sum\frac{1}{i}}{n})$

левая часть стремится к 0, а правая к x

ч.т.д. если условие внутри цикла выполнилось 1 раз, то оно и дальше внутри этого цикла будет выполнятся

значит это условие может не выполнится только в начале цикла, пока `inner` маленький

пусть наше условие не выполнится k раз, это значит что

$$\frac{y-k}{k} \geq x$$

$\rightarrow$

$k \leq \frac{y}{x+1}$

тогда условие не выполнится $\dfrac{y}{x+1}$ раз

давайте оценим, сколько на сколько повышается y в конце каждого цикла по `inner`

в этой строчке

```
y = y + outer / inner;
```

когда `inner` становится больше чем `outer` к y прибавляется 0

таким образом в конце цикла по `inner`

$y_{new} = y_0 + outer \sum_2^{outer} \dfrac{1}{i}$

как мы знаем из матанализа $\sum \frac{1}{i} \rightarrow lnn + \gamma$

(где γ постоянная эйлера)

тогда $y_{new} = y_0 + (outer)ln(outer) + \gamma$

тогда в i-тую итерацию цикла `outer`

$y_i = \sum 2^j ln 2^j = \frac{1}{log_2 e} \sum j 2^j =$

$$\frac{2(2^i i - 2^i + 1)}{log_2 e}$$

что косается x

в i-тую итерацию цикла `outer`, $x_i = 2^{i+1} + 99$

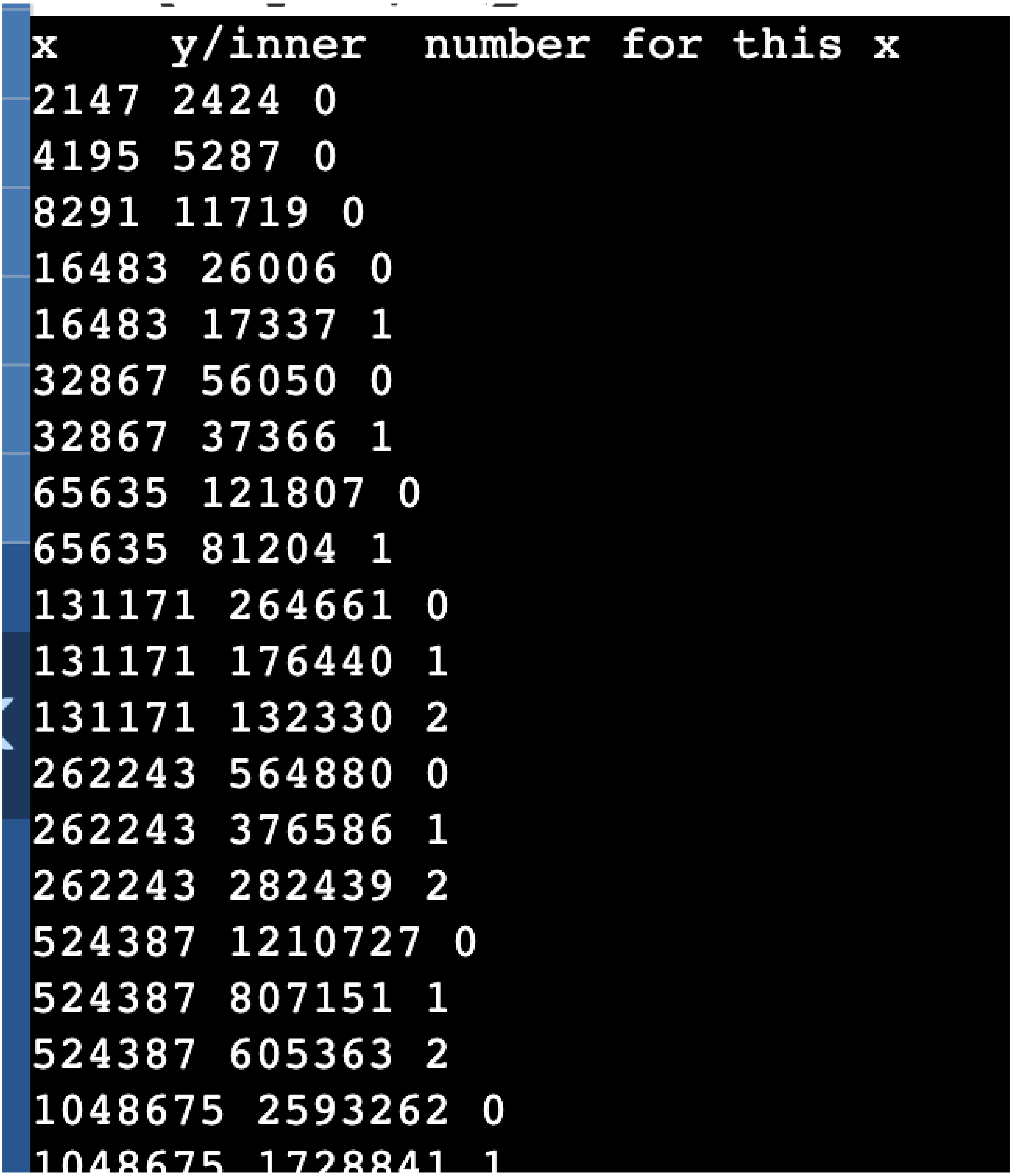
теперь мы можем найти k сколько раз условие не будет выполнятся:

$\frac{2^i(i-1)+2}{log_2 e(2^{i+1}+99)}$

при больших i стремится к

$$\frac{i}{2log_2e}$$

такми образом, условие не выполнится именно столько раз для каждого x невыполнится условие



```
x    y/inner  number for this x
2147 2424 0
4195 5287 0
8291 11719 0
16483 26006 0
16483 17337 1
32867 56050 0
32867 37366 1
65635 121807 0
65635 81204 1
131171 264661 0
131171 176440 1
131171 132330 2
262243 564880 0
262243 376586 1
262243 282439 2
524387 1210727 0
524387 807151 1
524387 605363 2
1048675 2593262 0
1048675 1728841 1
```

и как мы видим, действительно количество невыполнившихся увеличивается на 1 примерно каждые 4 итерации

теперь мы можем посчитать временную сложность

$$T(N) = 2 + log_2n(2 + 2 + n(2 + 2 + 3)) - 2(1 + 1 + 1 + 1 + 2 + 2 + 2 + 2 + 3 + ... + lnn/2) =$$

$$2 + log_2n(2 + 2 + n(2 + 2 + 3)) - 2\frac{4lnn/2(lnn/2 + 1)}{2} =$$

$$2 + log_2n(4 + 7n) - 2lnn^2 = 7nlog_2n + 4log_2n - 2lnn^2 + 2$$

$$T(N) = 7Nlog_2N + 4log_2N - 2ln^2N + 2$$

2. Найдите функцию $f(n)$, для которой справедливо соотношение $T(n) = \theta(f(n))$. Обоснуйте свой ответ в соответствии с определением θ–нотации.

$f(N) = NlogN$

докажем:

$7Nlog_2N + 4log_2N - 2(lnN)^2 + 2 < CNlog_2N$

работает с $C = 10$

$7 + \frac{4}{N} - 2\frac{lnN}{N} + \frac{2}{NlogN} < 10$

$\frac{4}{N} - 2\frac{lnN}{N} + \frac{2}{NlogN} < 3$

слева б.м. справа константа

проверим для изначальное условие для $N = 16$

$448 + 16 - 32 + 2 = 434 < 640$

докажем ограничение снизу:

$7Nlog_2N + 4log_2N - 2(lnN)^2 + 2 > CNlog_2N$

работает с $C = 1$

$6 > -\frac{4}{N} + 2\frac{lnN}{N} - \frac{2}{NlogN}$

слева константа, справа б.м. выполняется

покажем что работает начиная с $N = 16$ для изначального условия

$434 > 64$

ч.т.д.