Задача A3. Точная функция T(n) и порядок ее роста

Дан следующий фрагмент программного кода на языке C++, где n – это некоторая заранее заданная положительная целочисленная константа:

NESTED_LOOPS.cpp

```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. При составлении выражения для T(n) обратите особое внимание на условный оператор — количество элементарных операций в разных ветвях условия отличается.

Инициализации перед циклами:

int x = 100; – присваивание,1 операция.

int y = 0; – присваивание, 1 операция.

Инициализация внешнего цикла:

size_t outer = 1; – присваивание, 1 операция.

Внешний цикл выполняется k раз, где $k = \lfloor \log_2 n \rfloor + 1$, потому что на каждой итерации outer умножается на 2 до тех пор, пока outer <= n.

На каждой итерации внешнего цикла:

Проверка условия внешнего цикла: outer <= n; – сравнение, 1 операция.

Операция внутри внешнего цикла: x = x + outer; – сложение и присваивание, 2 операции.

Инициализация внутреннего цикла: size_t inner = 2; – присваивание, 1 операция.

Внутренний цикл выполняется n - 3 раз (от inner = 2 до inner = n - 1).

На каждой итерации внутреннего цикла:

Проверка условия внутреннего цикла: inner < n; – сравнение, 1 операция.

Условие if:
Вычисление y / inner; – деление, 1 операция.
Сравнение x > y / inner; – сравнение, 1 операция.

Ветвление:
Если условие истинно (true ветка):
- outer / inner; – деление, 1 операция.
- y + (результат); – сложение, 1 операция.
- y = (результат); – присваивание, 1 операция.

Если условие ложно (false ветка):
- (- -y); – декремент, 1 операция.

Инкремент внутреннего цикла: ++inner; – инкремент, 1 операция.

Итого на каждую итерацию внутреннего цикла:
Минимум: 1 (условие) + 2 (if) + 1 (ветка false) + 1 (инкремент) = 5 операций.
Максимум: 1 (условие) + 2 (if) + 3 (ветка true) + 1 (инкремент) = 7 операций.

Завершение внешнего цикла:

Инкремент внешнего цикла: outer *= 2; – умножение и присваивание, 2 операции.

Общие операции на каждой итерации внешнего цикла:

Проверка условия внешнего цикла: 1 операция.
Операции внутри внешнего цикла: 2 (x = x + outer) + 1 (инициализация внутреннего цикла) + 2 (инкремент outer) = 5 операций.

Операции во внутреннем цикле: (n - 3) * (количество операций на итерацию внутреннего цикла).

Общее количество операций:
1. Инициализации перед циклами: 2 операции.
2. Инициализация внешнего цикла: 1 операция.
3. Сумма по итерациям внешнего цикла:

$$\sum_{i=1}^{k}[1(\text{проверка условия внешнего цикла})+5(\text{операции внутри внешнего цикла})+$$
$$+(n-3)\times(5 \text{ или } 7 \text{ в зависимости от ветки})]$$

4. Проверка условия внешнего цикла при выходе из цикла: 1 операция.

Итого, общее количество операций T(n):

$$T(n) = 2 + 1 + (k+1) + k \times 5 + k \times (n-3) \times$$
$$\times(\text{среднее количество операций во внутреннем цикле})$$

Т.к точное количество операций зависит от того, сколько раз выполняются ветки true и false внутри условного оператора, можем положить, что T(n) пропорционально $n \times k$, где $k = \lfloor \log_2 n \rfloor + 1$.

2. Найдите функцию f(n), для которой справедливо T(n) = Θ(f(n)). Обоснуйте свой ответ в соответствии с определением Θ-нотации.

Из вывода в первом вопросе мы получили, что общее количество операций T(n) пропорционально (n * k) , где $k = \lfloor \log_2 n \rfloor + 1$. Таким образом:

$$T(n) = \Theta(n \log n)$$

Обоснование:

По определению Θ-нотации, функция T(n) принадлежит $\Theta(f(n))$, если существуют такие положительные константы c_1, c_2 и n_0, что для всех $n \geq n_0$ выполняется:

$$c_1 \cdot f(n) \leq T(n) \leq c_2 \cdot f(n)$$

В нашем случае $f(n) = n \log n$. Из вычислений следует, что T(n) ограничена сверху и снизу функциями, пропорциональными $n \log n$, что соответствует определению $\Theta(n \log n)$.