2415015T 山田　光樹矢
課題 0512

(1)
n = 10 のとき返り値 3
n = 50 のとき返り値 21
n = 100 のとき返り値 50

(2)

```
int pyth(int n) {
    int count = 0;
    for (int a = 1; a <= n; a++) {
        for (int b = 1; b * b < a * a; b++) {
            for (int c = b; b * b + 2 * c * c <= a * a; c++) {
                if (a * a == b * b + 2 * c * c) {
                    count++;
                }
            }
        }
    }
    return count;
}
```

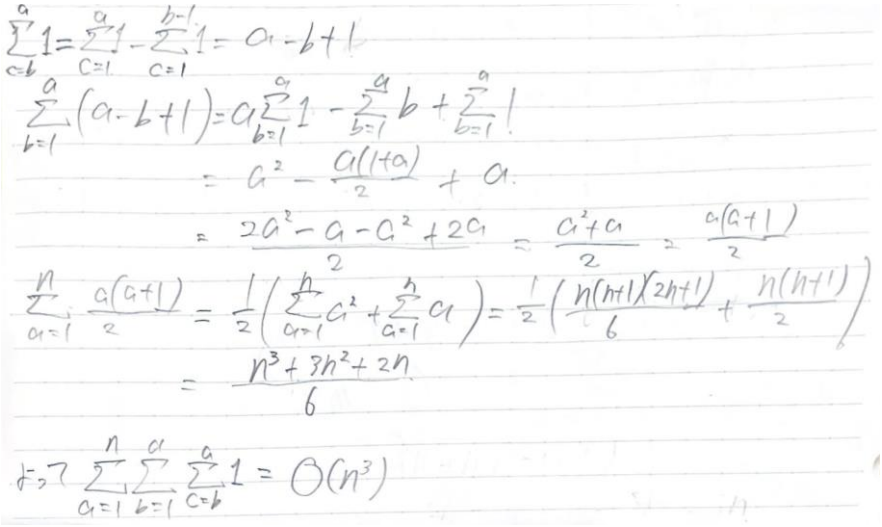
「処理が何回繰り返されるか」＝「計算時間がかかる量」と考えられるため、各 for 文が何回繰り返されるかを調べることで計算量が求められる。

・外側のループ： for (int a = 1; a <= n; a++)
回数：n 回

・中間のループ： for (int b = 1; b * b < a * a; b++)
条件 b^2 < a^2 より、おおよそ b < a となる
回数は約 a 回（最悪で n 回）

・内側のループ： for (int c = b; b^2 + 2c^2 <= a^2; c++)
条件を変形すると c^2 <= (a^2 - b^2) / 2
よって漸近的に考えると最大 c ≈ a なので、回数は高々 a 回
このことから関数 pyth(n)の時間計算量のオーダは次の式で表されると考えられる。

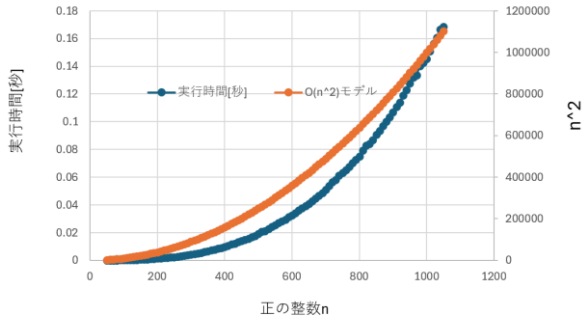
$\sum_{a=1}^{n} \sum_{b=1}^{a} \sum_{c=b}^{a} 1$



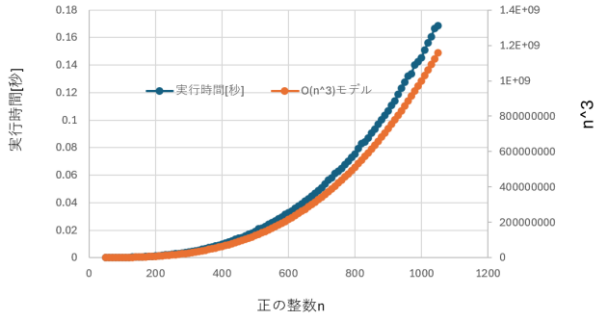
結論

pyth(n)の時間計算量は O(n^3)である。

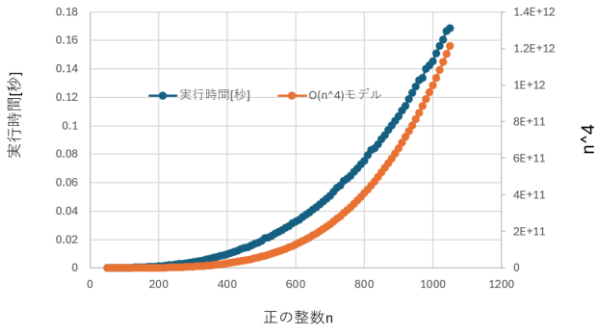
(3)



正の整数nと実行時間、n^2の関係



正の整数nと実行時間、n^3の関係



正の整数nと実行時間、n^4の関係

100 回繰り返してその平均をとったものを実行時間としている。
n を 50 から 1050 まで 10 ずつ変化させてプロットした。
考察
O(n^2)や O(n^4)と比べても O(n^3)は実行時間とおおむね一致しているため、O(3)と実際の処理時間は整合していると考えられる。
実行時間が理論的な計算量である O(n^3)より少し上にある部分が見られるが、これは CPU への負荷で合ったりキャッシュの影響などの実行環境によるオーバーヘッドが原因と考える。