计算物理作业 1

杨远青 22300190015

2024 年 9 月 13 日

1 题目 1：五次幂丢番图方程

1.1 题目描述

Find all integer solutions to the **Diophantine equation** $a^5 + b^5 + c^5 + d^5 = e^5$ within the range $[0, 200]$.

1.2 程序描述

该方程的第一个解在 1967 年被 Lander 和 Parkin 使用一台 CDC 6600 计算机找到，他们发现了 $27^5 + 84^5 + 110^5 + 133^5 = 144^5$。这个解同时还证伪了欧拉幂和猜想（即 k 次幂至少需要另外 k 个 k 次幂才能表示,k 大于等于 2)。参考 Diophantine Equation–5th Powers（https://mathworld.wolfram.com/DiophantineEquation5thPowers.html）第二个解是 J. Frye 在 2004 年使用分布式并行计算找到的，$55^5 + 3183^5 + 28969^5 + 85282^5 = 85359^5$ 题目要求解满足 $a^5 + b^5 + c^5 + d^5 = e^5$ 的整数解，其中 $0 \leq a \leq b \leq c \leq d < e \leq 200$。我们可以通过暴力搜索（brute-force）的方法来解决这个问题，即遍历所有可能的 a, b, c, d, e 组合，检查是否满足方程。在 $brute_force.90$ 中我们便使用了这个方案，伪代码见

但是这种方法的时间复杂度为 $O(N^5)$，其中 $N = 200$。我们可以通过一些技巧来减少搜索的时间复杂度。

1.3 伪代码

Algorithm 1: Brute-force solution to the Diophantine equation

Input: N: Integer (the upper bound, $N = 200$)

Output: *solutions*: List of tuples (a, b, c, d, e); // where $0 \leq a \leq b \leq c \leq d < e \leq N$

```
for a ← 0 to N do
    for b ← a to N do
        for c ← b to N do
            for d ← c to N do
                for e ← d + 1 to N do
                    if a^5 + b^5 + c^5 + d^5 = e^5;          // Check if the tuple is a solution
                     then
                        solutions.append((a, b, c, d, e)) ;          // Store the solution tuple
                    end
                end
            end
        end
    end
end
return solutions;                                          // Return the list of solution tuples
```

Algorithm 2: Mod30 trick for solving the Diophantine equation

Input: N: Integer (the upper bound, $N = 200$)

Output: *solutions*: List of tuples (a, b, c, d, e); // where $1 \leq a \leq b \leq c \leq d < e \leq N$

```
for a ← 1 to N do
    for b ← a to N do
        for c ← b to N do
            for d ← c to N do
                r_left ← mod(a + b + c + d, 30) ;          // Compute remainder for e
                for e ← d + mod(r_left − d, 30) to N, step 30 do
                    if a^5 + b^5 + c^5 + d^5 = e^5 then
                        solutions.append((a, b, c, d, e)) ;          // Store the solution tuple
                    end
                end
            end
        end
    end
end
return solutions;                                          // Return the list of solution tuples
```

Algorithm 3: Reverse Mod30 Trick for Solving the Diophantine Equation

Input: N: Integer (the upper bound, $N = 200$)

Output: *solutions*: List of tuples (a, b, c, d, e); // where $1 \leq a \leq b \leq c \leq d < e \leq N$

```
for e ← N to 1 do
    for d ← e to 1 do
        for c ← d to 1 do
            for b ← c to 1 do
                a_min ← mod(e − d − c − b, 30) ;      // Compute minimal a using modular arithmetic
                if a_min ≤ 0 then
                    a_min ← a_min + 30 ;              // Adjust a_min to fit within the modulus
                end
                for a ← a_min to b, step 30 do
                    if a^5 + b^5 + c^5 + d^5 = e^5 then
                        solutions.append((a, b, c, d, e)) ;      // Store the solution tuple
                    end
                end
            end
        end
    end
end
return solutions;                                     // Return the list of solution tuples
```

1.4 输入输出实例