УДК 004.424.36

Рубан М.М.

[1] *асистент кафедри алгебри, СДПУ*

e-mail: rubannn@gmail.com

ПРОБЛЕМА ВИКОРИСТАННЯ РЕКУРСІЇ НА ПРИКЛАДІ КОНКРЕТНОЇ ЗАДАЧІ

Розглядається один з методів пошуку рекурсивного розв'язку задачі та підхід який дозволяє зробити перехід від рекурсивного розв'язку для лінійного (математичного).

Ключові слова: *рекурсія, алгоритм, енциклопедія послідовностей цілих чисел*

Вступ

Дуже багато уваги приділяється задачам які мають рекурсивний розв'язок, але, як відомо, не завжди задачі на які легко знайти рекурсивний розв'язок повинні розв'язуватися саме цим методом. Загальновідомо, що рекурсія є методом який залежить від кількості пам'яті яку виділено для рішення задачі, тому завжди потрібно робити спроби знаходження аналогічного ітеративного або лінійного розв'язку. Розглянемо такі підходи на конкретній задачі.

В вершинах решітки $n \times n$ міститься n^2 точок (кожна комірка решітки — квадрат розміру 1×1). Скільки існує всіляких квадратів з вершинами в цих точках (сторони квадратів не обов'язково паралельні рядкам і стовпцям решітки)?

Основна частина

За допомогою рисунку (рис. 1) визначимо кількість різних квадратів при $n = 4$. Легко бачити, що квадратів буде рівно $1 + 4 + 9 + 4 + 2 = 20$ штук.

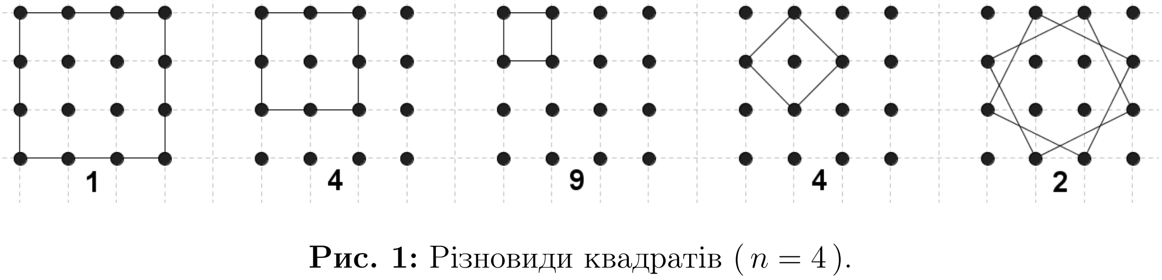

Рис. 1: Різновиди квадратів ($n = 4$).

Знайдемо загальний розв'язок. Позначимо через $S(k)$ загальну кількість різних квадратів які можна побудувати на решітці $k \times k$, де k — довжина зовнішньої границі решітки, тобто решітка складається з $(k+1)^2$ точок.

Зрозуміло, що $S(1) = 1$, тобто лише один квадрат можна побудувати на решітці розміром 1×1. Для того щоб знайти значення $S(k+1)$ помітимо, що при розширенні решітки розміру $k \times k$ до розміру $k+1 \times k+1$ додатково утворюється рівно $(2k+1)$ квадрат розміром 1×1, $(2k-1)$ квадрат розміром 2×2, $(2k-3)$ квадратів розміром 3×3, ..., 3 квадрати $k \times k$ та один квадрат $k+1 \times k+1$. Таким чином отримаємо наступну рекурентну формулу:

$$S(k+1) = S(k) + (2k+1) + (2k-1) + (2k-3) + \cdots + 3 + 1$$

$$S(k+1) = S(k) + \sum_{i=0}^{k} (2i+1) \tag{1}$$

Але при знаходженні формули (1) була врахована лише кількість різноманітних квадратів сторони яких паралельні решітці.

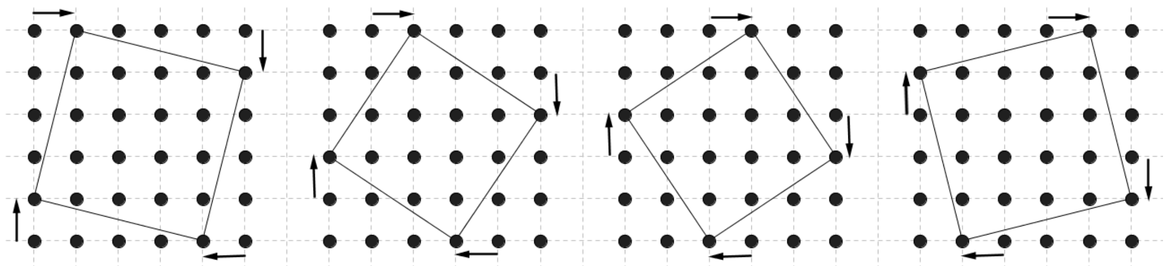

Рис. 2: Отримання r-квадратів шляхом обертання ($n = 6$).

Для знаходження повної формули, потрібно додатково знайти кількість різноманітних квадратів сторони яких не паралельні решітці (назвемо їх r-квадратами). Аналогічно попереднім міркуванням знайдемо кількість r-квадратів (відповідних розімірів) які утворюються при переході від решітки $k \times k$ до $k+1 \times k+1$. Помітимо, що r-квадрати утворюються під час одночасного зміщення всіх вершин в одному напрямку на одну точку (рис. 2 — r-квадрати які отримані для решітки 6×6). Неважко впевнитись що загальну кількість r-квадратів, яка отримується при переході від решітки $k \times k$ до $k+1 \times k+1$, рівна:

$$\sum_{i=0}^{k} (2i+1)(k-i).$$

Таким чином, для знаходження загальної кількості квадратів слід використовувати таку формулу:

$$\begin{aligned} S(k+1) &= S(k) + \sum_{i=0}^{k}(2i+1) + \sum_{i=0}^{k}(2i+1)(k-i) = \\ &= S(k) + \sum_{i=0}^{k}\left((2i+1) + (2i+1)\,(k-i)\right) = \\ &= S(k) + \sum_{i=0}^{k}(2i+1)\,(k-i+1) = \\ &= S(k) + \frac{(k+1)(k+2)(2k+3)}{6}. \end{aligned} \tag{2}$$

Або в більш зручній формі: $S(k) = S(k-1) + \dfrac{(k+1)k(2k+1)}{6}$, $(*)$
де k — довжина зовнішньої границі решітки.

Програмна реалізація даного рішення на мові *Pascal*:

```
function sq(k:int64):int64;
begin
  if (k = 1)
  then sq:=1
  else sq:=sq(k-1)+k*(k+1)*(2*k+1) div 6;
end;
var n:int64;
begin
  read(n);
  writeln(sq(n));
end.
```

Але наведена програма при обчисленні значення для $n \geq 7802$ виведе помилку — переповнення стеку під час обчислення рекурсії, тому слід знайти інший нерекурсивний розв'язок. Для цього звернемо увагу на формулу $(*)$ — для знаходження кожного наступного значення $S(k)$ використовується вже обчислене значення $S(k-1)$, тому якщо при підрахунку зберігати попереднє $S(k-1)$, то легко отримати нерекурсивний алгоритм.

```
var n,s,s0,i:int64;
begin
  read(n);
  s:=0; s0:=0; i:=1;
  while (i <= n) do
```

```
  begin
    s:=s0 + i*(i+1)*(2*i+1) div 6;
    s0:=s;
    i:=i+1;
  end;
  writeln(s);
end.
```

Даний розв'язок вже позбавлений вище описаного обмеження яке стосується переповнення стеку (виключаючи випадки переповнення типу *int64*), тому саме його бажано використовувати як основний.

Розглянувши додатково ряд відповідей (1, 6, 20, 50, 105, 196, ...), які отримуються під час обчислень, та використавши online енциклопедію послідовностей цілих чисел (*The On-Line Encyclopedia of Integer Sequences*) можна знайти назву цієї послідовності — пірамідальні числа 4 порядку (*4-dimensional pyramidal numbers*) та формулу загального елемента $\dfrac{C_{n^2}^2}{6}$ або $\dfrac{n^2(n^2-1)}{12}$. Використовуючи останню формулу легко отримати лінійний алгоритм розв'язку.

Висновки

Отримані результати підкреслюють той факт, що за допомогою рекурсії велика кількість задач розв'язується легко та красиво, як кажуть «в один рядок». Але під час використання рекурсії досить часто виникають проблеми при її використанні, як було наведено на прикладі, тому в деяких випадках слід прораховувати такі варіанти, та у разі, якщо рекурсивний розв'язок може привести до хибних або помилкових результатів, то краще використати його ітеративний аналог або взагалі, якщо це можливо, лінійний.

Література

[1] *Ахо А.* Структуры данных и алгоритмы : [пер. с англ.] / А. Ахо, Дж. Хопкрофт, Дж. Ульман – Москва: Вильямс, 2007. – 400 с.

[2] *Грэхем Р.* Конкретная математика. Основы информатики / Р. Грэхем, Д. Кнут, О. Паташник – М.: Мир; БИНОМ. Лаборатория знаний, 2006. – 703 с.

[3] *Шень А.* Программирование: теоремы и задачи / А. Шень. – 2-е изд., испр. и доп. – М.: МЦНМО, 2004. – 296 с.