根號算法 1

1.1 前言

「十則圍之、五則攻之、倍則分之」，這是孫子兵法中提到的，說要視面對敵人的多寡，要採用不同的策略。同樣地，如果題目給的範圍有不同的限制，可能需要不同的作法。有時，這個限制可可能會在題目中就給了；不過，有時候題目連這個都不會給！所以，就必須要自己來界定分離的標準了。這個方法有時候看起來很通靈，甚至很唬爛，但是經過分析之後是對的，時常能將複雜度的一個變數降為其根號，譬如一個 $O(N^2)$ 的演算法其實經過一點巧思是 $O(N\sqrt{N})$，雖然會比 log 等級的演算法慢一些，但是通常實作複雜度低許多，是一個很實用的工具！

1.2 種類壓大小

還是一樣，先看一個簡單的例題吧：

習題 1.2.1: 詭異的詢問（No Judge）

我有一個神秘函數 $f(x)$。每次，你都可以詢問一個 x，我就會回你 $f(x)$ 的值。另外一個人會問你 $Q \leq 10^6$ 個正整數 k_i，代表你要回答說 $f(k)$ 等於多少，而且保證 $\sum_{i=0}^{Q-1} k_i \leq 10^8$。然而，我很不耐煩，所以你只能問我 1000 次的詢問。請好好利用這 15000 次的詢問，來回答另外一個人的 10^6 個詢問吧！

乍看之下，這題根本不可做——有 10^6 筆詢問，可能有 10^8 個不同的數字，而我只有 10^4 次的詢問，根本不可做！首先，會先想到說如果重複詢問的話，那就先把問過的東西存下來再回答就好啦！所以呢，現在的問題就變成：到底有多少個可能的詢問呢？不過，或許眼尖的你注意到了奇怪的限制：$\sum_{i=0}^{Q-1} k_i \leq 10^8$！這有什麼用呢？想要讓數字最多種，當然要讓詢問的數字越小越好，才比較不會超越所給的限制，才會有最多種的數字呀！所以，假設是有 n 種數字，則一定是依

序詢問 $1, 2, 3, \ldots, n$ 才好，所以呢：

$$\begin{aligned} \frac{n(n+1)}{2} &\leq 10^8 \\ (n+1)^2 &\leq 2 \times 10^8 \\ n &\leq \sqrt{2 \times 10^8} - 1 \approx 14141 < 15000 \end{aligned}$$

也就是呢，最多有 14141 種相異個數字會出現，所以只要存下來就好了！運用這個技巧，可以推論出一個結論：

定理 1.2.1: 和與種類的不等式

假設有一堆數字的和為 S，則那些數字的種類數量的數量級為 $O(\sqrt{S})$。

此定理的證明和上面是一樣的，這裏就不贅述了。

1.3 不會有那麼多個吧！

在一個圖中尋找三角形是很經典的一個問題：

習題 1.3.1: 尋找三角形（經典問題）

給定一個有 $N \leq 10^4$ 個點，$M \leq 10^4$ 個邊的圖，請問這個圖有幾個三角形？一個三角形的定義為一個無序的 3-tuple (u, v, w)（$1 \leq u, v, w \leq N$），使得那三個點兩兩有邊連接。

先直接丟唬爛的解：枚舉一個邊 (u, v)，看 $\deg(u)$ 和 $\deg(v)$ 哪一個比較小 (假設是 u)，那就直接硬枚舉所有與 u 連結的點 x，並看 x 和 v 是否有沒有連結，如果有的話，就直接加一。這樣，每個三角形都會被數到三次，所以就輸出答案除以三就好了！這樣的複雜度是 $O(NM)$，因為對於每一個邊，都有可能掃到每另外一個邊。