Sprout 2020 Algorithm - Week 6

Author: 陳楚融

Problem 1

(a)

先考慮時間，第 i 次擴張需複製 2^{i-1} 個元素，令 $2^k = 2^{\lceil \log_2 n \rceil}$ ，即最後陣列大小，最後一次擴張共複製 2^{k-1} 次，共 $S = \sum_{i=1}^{k} 2^{i-1} = 2^k - 1$ 次， $n - 1 \leq S < 2n$ 加上插入共 n 個元素，最後時間複雜度為 $O(2n + n) = O(n)$ 。再考慮空間， 第 i 次擴張最多同時存在 $2^i + 2^{i-1}$ 個空間，最多 $\frac{3}{2} * 2^k < \frac{3}{2} * 2n = 3n$ ，得空間複雜度為 $O(n)$

(b)

先考慮時間，第 i 次擴張需複製 i 個元素，最後一次擴張共複製 $n - 1$ 次，共 $S = \sum_{i=1}^{n-1} i = \frac{n^2 - n}{2}$ 次，加上插入共 n 個元素，最後時間複雜度為 $O(\frac{n^2 + n}{2}) = O(n^2)$ 。再考慮空間， 第 i 次擴張最多同時存在 $i + (i + 1)$ 個空間，最多 $(n - 1) + n = 2n - 1$ ，得空間複雜度為 $O(n)$

Problem 2

(a)

抵達環的起點（入口） u 後，剛好走其長度 L 步會回到 u ，其他節點都恰好走過一次，因此可在每個節點裡宣告一變數 t ，紀錄已走步數，如 $start \to t = 0$ ，則第二次走到 u 時的總步數減去 $u \to t$ 即為 L 。除 u 外所有 $n - 1$ 個節點恰好走過一次，u 走過兩次，因此時間複雜度為 $O(n)$

(b)

由起點開始走 k 步會進入環，並用變數 p 紀錄位置，再用從 p 開始走 $k - 1$ 步，由於已在環內，因此若環長 $L < k$ ，必定會再回到 p ，若回到 p 則 L 為走過步數，否則 $L \geq k$ 。總共使用一個變數 p 與一個變數用來遍歷圖，故空間複雜度為 $O(1)$ 。從起點走 k 步，再從 p 走 $k - 1$ 步，得時間複雜度 $O(2k - 1) = O(k)$

(c)

由 $L \leq n$ 可知 $\log_2 n \notin \mathbb{N}$ 時， k 最多為 $2^{\lceil \log_2 n \rceil} > n$ ， $\log_2 n \in \mathbb{N}$ 時，k 最多為 $2^{1 + \log_2 n} > n$ ，因為由起點走 k 步後會進入環，再走 $k - 1$ 步至少會繞環一圈並得到 L 。每次判斷皆需 $O(k)$ ，且 $\log_2 n \notin \mathbb{N}$ 時， $\sum_{i=0}^{\lceil \log_2 n \rceil} 2^i = 2^{\lceil \log_2 n \rceil + 1} - 1 < 2 * 2n - 1 < 4n$ ， $\log_2 n \in \mathbb{N}$ 時， $\sum_{i=0}^{1 + \log_2 n} 2^i = 2^{2 + \log_2 n} - 1 < 4 * n - 1 < 4n$ ，得時間複雜度為 $O(n)$ 。判斷時使用之變數仍為兩個，加上一個 k ，故空間複雜度為 $O(1)$

Problem 3

(a)

若 a_i 皆為 1，每次最多跳 $N-1$ 次，共 Q 次，得時間複雜度為 $O(NQ)$

(b)

$a[a[i]]$

(c)

$b[b[i][j-1]][j-1]$

(d)

對 k 做二分拆解（轉成二進位），得 $k=2^{p_1}+2^{p_2}+\cdots+2^{p_q}$， $p_q \le \lfloor \log_2 k \rfloor$ ，複雜度為 $O(log\ k)$ ，然後從 x 開始依序跳 p_1，p_2，$\cdots$，p_q 步，共 k 步，每次可透過陣列 b 使用 $O(1)$ 的時間完成，最多跳 $\lfloor \log_2 k \rfloor$ 次，又因 $k<N$ ，得包含拆解的總時間複雜度為 $O(log\ N)$

(e)

建立陣列 b 時， $0 \le j \le \lfloor \log_2 N \rfloor$ ，否則會跳超過 N 步，因此陣列 b 元素數 $B \le N(1+\log_2 N)$ ，每個元素計算時間僅 $O(1)$ ，得建表時間複雜度為 $O(N\ log\ N)$。每筆詢問計算時間為 $O(log\ N)$， Q 筆詢問共 $O(Q\ log\ N)$ ，得總時間複雜度為 $O((N+Q)\ log\ N)$