

W6 手寫功課

劉至軒

May 18, 2019

Problem 1.

1. 一個東西插入會是 $O(1)$ ，如果不考慮長度變長的話就是 $O(N)$ ；現在來考慮長度變長的 case：會是

$$\sum_{k=0}^{\lfloor \log N \rfloor} 2^k = 2^{\lfloor \log N \rfloor + 1} - 1 \approx 2N - 1 = O(N)$$

，加起來時間複雜度是 $O(N)$ 。而空間複雜度：

$$2^{\lfloor \log N \rfloor} \leq 2N = O(N)$$

2. 顯然的，空間複雜度是 $O(N)$ ，因為 `capacity` 每次只會加一，所以會是 $\min(C_0, N)$ ， C_0 是初始的 `capacity`。至於時間複雜度，最差的 case 就是一開始的 `capacity` 是零，所以每次加都要重新 copy 一次，複雜度

$$\sum_{k=0}^n k = \frac{n(n-1)}{2} = O(n^2)$$

Problem 2.

1. 在每一個 `Node` 裡面增加一個變數 `time`，代表進來的時間戳，一開始全部都是 0，一開始時間為 1，每次去 `next` 就判斷其 `time` 是否為 0，如果是的話就設定為 `++currentTime`，否則回傳換的大小為 `currentTime - next->time`。這樣， n 個 `Node` 都增加了一個變數，所以空間複雜度為 $O(n)$ 。
2. 可以知道，假設環和起點的距離為 r ，環長度為 λ ，則對於 $t \geq r$ ，和所有的 $x \in \mathbb{N}$ ，有

$$x_t = x_{t+\lambda x}$$

，也就是

$$a \equiv b \pmod{\lambda} \implies x_a = x_b$$

最差情況就是我跳了 λ 步，則顯然 $\lambda \equiv 2\lambda \pmod{\lambda}$ ，而因為 $2\lambda - \lambda = \lambda$ ，所以就直接輸出答案即可。在這之前，都不會有 $0 < k < \lambda$ 使得 $x_k = x_{2k}$ ，因為如果 $k \equiv 2k \pmod{\lambda} \implies \lambda | (2k - k) = k$ ，但是 $k < \lambda \implies k = 0$ ，然而 $k > 0$ ，所以不會有。如果 $\lambda \leq k$ ，則就最多跑 k 次，如果有遇到 $x_i = x_{2i}$ 的話，那環的大小就是 i ；否則就輸出沒有環。這樣複雜度就是 $O(k)$ 了。

3. 令環長為 $\lambda = O(N)$ ，則會經過 $\lceil \log \lambda \rceil$ 次的尋找才會找到。所以時間複雜度

$$\sum_{k=0}^{\lceil \log \lambda \rceil} 2^k = 2^{\lceil \log \lambda \rceil + 1} - 1 = O(\lambda) = O(N)$$

Problem 3.

1. 最爛的 case 就是每次 a_i 都要跳到 a_{i+1} ，所以要慢慢跳，跳 $O(N)$ 格，而如果每一個詢問都這樣整人的話，複雜度就是 $O(NQ)$ 了。
2. 可以表示為 $a[a[i]]$ 。
3. 應該要填 $b[b[i][j-1]][j-1]$ 。
4. 令 $q = \lceil \log k \rceil$ ，那以下的程式會回傳一開始在 x ，跳了 k 步之後的編號：

```

1  for(int i = 0; i <= q; i++){
2      if((long long)1 << i & k){
3          x = b[x][i];
4      }
5  }
6  return x;

```

所以複雜度是 $O(\log k)$ 。

5. 因為已經假設好 b 陣列已經建立完畢（如果沒有的話可以 $O(N \log N)$ 建立），對於每一個詢問都可以 $O(\log k) = O(\log N)$ 詢問，所以複雜度 $O(Q \log N)$ 。