Sprout 2020 Algorithm - Week 10

Author: 陳楚融

Problem 1

設 $\text{lowbit}(x) = 2^k$ ，x 之位元由小至大依序編號為 $b_0^x, b_1^x, b_2^x, \cdots, b_n^x$

令 $0 \le i < k < j \le n$ ：

$$
\begin{aligned}
& b_i^x = 0,\ b_k^x = 1,\ 0 \le i < k \\
\implies & b_i^{\neg x} = 1,\ b_k^{\neg x} = 0,\ b_j^{\neg x} = \neg b_j^x \\
\implies & b_i^{\neg x+1} = 0,\ b_k^{\neg x+1} = 1,\ b_j^{\neg x+1} = \neg b_j^x \\
\implies & x \& (\neg x + 1) = 2^k
\end{aligned}
$$

得證 $\text{lowbit}(x) = x \& (\neg x + 1)$

Problem 2

a

$$
\begin{aligned}
& \sum_{i=0}^{n} i * 2^{i-1} = \sum_{i=1}^{n} i * 2^{i-1} = \sum_{i=1}^{n} 2^{i-1} + \sum_{i=2}^{n} (i-1) * 2^{i-1} \\
= & \sum_{i=1}^{n} 2^{i-1} + \sum_{i=2}^{n} 2^{i-1} + \sum_{i=3}^{n} (i-2) * 2^{i-1} = \cdots \\
= & \sum_{i=1}^{n} \sum_{k=i}^{n} 2^{k-1} = \sum_{i=1}^{n} \frac{2^{i-1}(2^{n-i+1} - 1)}{2 - 1} = \sum_{i=1}^{n} 2^n - 2^{i-1} \\
= & n * 2^n - \sum_{i=0}^{n-1} 2^i = n * 2^n - \frac{1(2^n - 1)}{2 - 1} = (n-1) * 2^n + 1
\end{aligned}
$$

b

已知 $m = 0$ 時命題成立 $f(0) = 1 \ge 1 * 2^0 = 1$

證明若 $n = k \ge 0$ 時命題成立，則 $n = k + 1$ 時命題也會成立：

$$
\begin{aligned}
& f(k+1) = 2^{k+1} + \sum_{i=0}^{k} f(i) \\
\ge & 2^{k+1} + \sum_{i=0}^{k} i * 2^{i-1} = 2^{k+1} + (k-1) * 2^k + 1 \\
= & (k+1) * 2^k + 1 \ge (k+1) * 2^{(k+1)-1}
\end{aligned}
$$

命題仍然成立

得證 $f(m) \ge m * 2^{m-1},\ \text{for } m \ge 0$

Problem 3

a

$[2, n]$

b

依據遞迴式得 $a = 2 \le b$ 時：

觀察結構圖可發現僅 $b = 2^k,\ k > 0$ 時， $\text{range}(b)$ 左界小於 a ，其餘情況皆大於 a

當 $b = 2^k,\ k > 0$ ，遞迴解 $\text{query}(a, b) = \max(\text{arr}[b], \text{query}(a, b-1))$;

否則遞迴解 $\text{query}(a, b) = \max(\text{bit}[b], b - \text{lowbit}(b))$

因此當 $b > 2^h = 2^{\lfloor \log n \rfloor}$ 時，最多遞迴 h 次會減至 2^h

接著每當 $b = 2^k,\ k \in \mathbb{N} - \{1\}$ 時，先遞迴至 $2^k - 1$ ，然後遞迴 $k-1$ 次至 2^{k-1} ，共 k 次

最後 $b = a$ 時，遞迴一次得 $\text{query}(a, b) = \max(\text{arr}[b], \text{query}(a, a-1)) = \text{arr}[b]$

總操作 $f(n)$：

$$
\begin{gathered}
1 + \sum_{i=2}^{h} i \le f(n) \le h + 1 + \sum_{i=2}^{h} i \\
\frac{h^2 + h}{2} \le f(n) \le \frac{h^2 + 3h}{2} \\
\implies f(n) \in O(\log^2 n)
\end{gathered}
$$

得此組詢問時間複雜度為 $O((\log n)^2)$

Problem 4

a

$\text{query}(\text{dif}, x)$

$$\text{query}(\text{dif}, x) = \text{arr}[1] + \sum_{i=2}^{x} \text{arr}[i] - \text{arr}[i-1] = \text{arr}[x]$$

b

$\text{update}(\text{dif2}, a, a * \text{val})$

$\text{update}(\text{dif2}, b+1, -(b+1) * \text{val})$

$\text{update}(\text{dif}, a, \text{val})$

$\text{update}(\text{dif}, b+1, -\text{val})$

若修改後陣列分別為 $\text{dif'}, \text{dif2'}$

對於 $x \in [a+1, b]$ ，更新後：

$$\begin{cases} \text{dif'}[x] &= (\text{arr}[x] + \text{val}) - (\text{arr}[x-1] + \text{val}) \\ &= \text{dif}[x] \\ \text{dif2'}[x] &= x * \text{dif'}[x] \\ &= \text{dif2}[x] \end{cases}$$

對於 a, b ，更新後：

$$\begin{cases} \text{dif'}[a] &= (\text{arr}[a] + \text{val}) - \text{arr}[a-1] \\ &= \text{dif}[a] + \text{val} \\ \text{dif2'}[a] &= a * (\text{dif}[a] + \text{val}) \\ &= \text{dif2}[a] + a * \text{val} \end{cases}$$

$$\begin{cases} \text{dif'}[b+1] &= \text{arr}[b+1] - (\text{arr}[b] + \text{val}) \\ &= \text{dif}[b+1] - \text{val} \\ \text{dif2'}[b+1] &= (b+1) * (\text{dif}[b+1] + \text{val}) \\ &= \text{dif2}[b+1] - (b+1) * \text{val} \end{cases}$$

c

根據 Problem4.a， $\sum_{i=1}^{x} \text{arr}[i] = \sum_{i=1}^{x} \sum_{k=1}^{i} \text{dif}[k]$

d

$(x+1) * \text{query}(\text{dif}, x) - \text{query}(\text{dif2}, x)$

$$\begin{aligned} \sum_{i=1}^{x} \text{arr}[i] &= \sum_{i=1}^{x} \sum_{k=1}^{i} \text{dif}[k] = \sum_{i=1}^{x} (x+1-i) * \text{dif}[i] \\ &= (x+1) * \sum_{i=1}^{x} \text{dif}[i] - \sum_{i=1}^{x} i * \text{dif}[i] \\ &= (x+1) * \text{query}(\text{dif}, x) - \text{query}(\text{dif2}, x) \end{aligned}$$