Binary Indexed Trees (Fenwick Trees)

Ulises Tirado Zatarain [1]
(ulises.tirado@cimat.mx)

[1]Algorists Group

August, 2015

Outline

1. Introduction
   - Basic Problem
   - Basic Idea
2. Operations
   - The last bit
   - Reading
   - Writing
   - Get single
3. Problems
4. Multidimensional BITs

Introduction - Basic problem

- We have n boxes and we need to perform two operations:
  1. Add marbles to box k: $\mathcal{O}(1)$
  2. Sum marbles from box i to j: $\mathcal{O}(n)$
- Suppose we make q queries. The worst case has time complexity $\mathcal{O}(qn)$.
- Using Segment Trees/RMQ: $\mathcal{O}(q \lg n)$.

Introduction - Basic idea

- Taking the binary representation of an integer k, we can compute the sum of elements from 1 to k as sum of set of sums.
- Each set contains son successive number of non-overlapping sums. Then, we can build a data structure indexed by the binary representation:

Binary Indexed Tree

Let k is an index of our BIT and r is the position in k of the last bit 1, then t_k is sum of boxes from index $k-2^r+1$ to index k. In other words, we mean that k is responsible for indexes in $[k-2^r+1, k] \cap \mathbb{Z}$.

CIMAT

Introduction - Basic idea

Responsibility table

k	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
R	1	1..2	3	1..4	5	5..6	7	1..8	9	9..10	11	9..12	13	13..14	15	1..16
v_k	1	0	2	1	1	3	0	4	2	5	2	2	3	1	0	2
t_k	1	1	2	4	1	4	0	12	2	7	2	11	3	4	0	29
Σ_k	1	1	3	4	5	8	8	12	14	19	21	23	26	27	27	29

- For example, if we are looking the sum of the first 13 elements, 13 is equal to 1101_2. Then, we will compute:

$$\Sigma_{1101_2} = t_{1101_2} + t_{1100_2} + t_{1000_2}$$

CIMAT

Isolating the last bit 1

- Let $x \in \mathbb{Z}$, then we can write x in binary representation as $[x]_2 = y1z$, where z consists of all zeros, so $\tilde{z}$ consists of al ones. Then, we have:

$$
\begin{aligned}
[-x]_2 &= \widetilde{y1z} + 1 \\
&= \tilde{y}0\tilde{z} + 1 \\
&= \tilde{y}01\ldots1 + 1 \\
&= \tilde{y}10\ldots0 \\
&= \tilde{y}1z
\end{aligned}
$$

- Using bitwise operator AND between x and $-x$, we can get:

$$
\begin{array}{cccc}
 & y & 1 & z \\
\& & \tilde{y} & 1 & z \\
\hline
= & 0\ldots0 & 1 & 0\ldots0
\end{array}
$$

CIMAT

Reading Σ_k: $\mathcal{O}(\lg n)$

- If we need the sum for some integer k, we add to sum t_k, substractnlas bit k from itself and repeat this until $k \leqslant 0$.

function $\mathtt{read}(k \in \mathbb{Z}^+)$ **begin**

    $sum \longleftarrow 0$;

    while $k > 0$ **do**

        $sum \longleftarrow sum + t_k$;

        $k \longleftarrow k - (k \ \& \ -k)$;

    end

    return sum;

end

Writing t_k: $\mathcal{O}(\lg n)$

- When we want to update the k-th element we need to update all indexes whose responsible for the element k.

function $\mathrm{add}(k \in \mathbb{Z}^+, value \in \mathbb{Z})$ **begin**
  while $k \leqslant n$ **do**
    $t_k \longleftarrow t_k + value$;
    $k \longleftarrow k + (k \ \& \ -k)$;
  end
end

Get single v_k: $\mathcal{O}(\lg n)$

function get$(k \in \mathbb{Z}^+)$ **begin**
    $sum \longleftarrow t_k$;
    if $k > 0$ **then:**
        $z \longleftarrow k - (k \ \& \ -k)$;
        while $k \neq z$ **do**
            $sum \longleftarrow sum - t_k$;
            $k \longleftarrow k - (k \ \& \ -k)$;
        end
    end
    return sum;
end

Problems

- There is an array of n cards. Each card is putted face down on table. You have two queries:
  1. $T(i,j)$ turn cards from index i to index j, include i-th and j-th card – card which was face down will be face up; card which was face up will be face down
  2. $Q(i)$ answer 0 if i-th card is face down else answer 1
- Floating Median

CIMAT

2D BITs $\mathcal{O}(q \lg m \lg n)$

function $\mathtt{add}(x, y \in \mathbb{Z}^+, value \in \mathbb{Z})$ **begin**
    $sum \longleftarrow 0$;
    while $x \leqslant n$ **do**
        $\hat{y} \longleftarrow y$;
        while $\hat{y} \leqslant m$ **do**
            $t_{x,\hat{y}} \longleftarrow t_{x,\hat{y}} + value$;
            $\hat{y} \longleftarrow \hat{y} + (\hat{y}\ \&\ -\hat{y})$;
        end
        $x \longleftarrow x + (x\ \&\ -x)$;
    end
end

CIMAT

References I

- TopCoder
- Math Porn @ tumblr
- Multidimensional BITs