

Lecture 7

Sunday, 19 January 2020 2:19 PM

Bit Masking, Manipulation.

1 \rightarrow 001
 2 \rightarrow 010
 3 \rightarrow 011
 4 \rightarrow 100
 5 \rightarrow 101
 6 \rightarrow 110
 7 \rightarrow 111
 8 \rightarrow 1000

2 \rightarrow 010
 1 \rightarrow 001
 3 \leftarrow 011

1
 010
 001
 ———
 000

$\left\{ \begin{array}{l} \text{syso}(2 \wedge 1) \rightarrow 3 \\ \text{syso}(2 \wedge 1) \rightarrow 0 \\ \text{syso}(2 | 1) \rightarrow 3 \\ \text{syso}(\sim 2) \rightarrow 5 \end{array} \right.$

010
 \rightarrow 101 5.

010
 001
 ———
 011

0 0 0 0 0 0 1 0

$$\begin{array}{r}
 (\rightarrow \\
 + \begin{array}{r}
 \begin{array}{cccccccc}
 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 \\
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0
 \end{array} \\
 \hline
 \begin{array}{cccccccc}
 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1
 \end{array}
 \end{array}
 \end{array}$$

```
int num = 16;  
num = num >> 2 (bits).
```

$\rightarrow \boxed{0} _ _ _ _ _ _ _ \boxed{0} = 2$

↳ 1st bit right shift → 00000001
↳ 00000000

16 \rightarrow 0 0 0 1 0 0 $\left\{ \begin{array}{c} \text{0} \\ \text{0} \end{array} \right\} \times$

○ ○ ○ ○ ○ | ○ ○
— — — — — — — — → (4)

010

$$16 / \sim 2^2 \Rightarrow (4)$$

~~17~~ 15

$$16 \ll 2 \Rightarrow (64)$$

$\begin{array}{ccccccc} 6 & 5 & 4 & 3 & 2 & 1 & 0 \\ \underline{0} & \underline{1} & \underline{0} & \underline{0} & \underline{0} & \underline{0} & \underline{0} \end{array} \rightsquigarrow 64.$

left shift \rightarrow multiply
 right shift \rightarrow divide.

$$\text{Syso}(\sim(2^i - 1)) \rightarrow \begin{array}{ccc} & 0 & 1 & 1 \\ \hline & 1 & 0 & 0 \end{array}$$

$\text{public static int get-bit}(\text{int num}, \text{int } i) \{$
 $\text{int mask} = 1 \quad // \begin{array}{cccc} & 10 & & \\ & 0 & 0 & 0 & 1 \end{array}$
 $\text{mask} \leftarrow \text{mask} \ll (i-1); \quad // \begin{array}{cccc} & 4 & & \\ & 0 & 0 & 1 & 0 \end{array}$
 $// \text{int ans} = \text{mask} \& \text{num};$
 $\text{return}(\text{mask} \& \text{num}) \neq 0 ? 1 : 0;$
 $\}$

$\begin{array}{cccc} & 10 & & \\ & 0 & 0 & 1 & 0 \\ & \leftarrow & & \\ & 0 & 1 & 0 & 0 \\ & \leftarrow & & \\ & 0 & 1 & 0 & 0 & \text{num} \\ & 1 & 0 & 0 & 0 & \text{mask} \end{array}$

~

```
public static void set_bit (int num, int i) {
```

mask = 1

mask = mask << i-1;

num = num | mask;

Syso(num);

4 3 2 1
 1 0 1 0 num
 0 1 0 0 mask
 3 2 1 0
 1 1 1 0
 8 + 4 + 2 = 14

4

```
public static int right_most_set_bit (int num) {
```

int pos = 1;

int mask = 1;

while ((num & mask) == 0) {

mask = mask << 1;

pos++;

}

return pos;

(2)

3 2 1
 1 1 0
 0 0 1
 0 0 0
 1 1 1
 0 0 0 1
 0 0 0 1

3

/)

0 1 0 mask.
 1 1 0

 0 1 0

public static boolean power of two (int n) {

if ((n & n-1) == 0) {

return true

}

return false;

}

16 →

1 0 0 0 0

 0 1 1 1 1

0 0 0 0 0

 0 0 0 0 0

0 1 1 1 1

0 1 1 1 0

0 1 1 1 0

0 0 0 0 0

(1) 0 0 0 1

int N = 9

arr = { 3, 4, 1, 5, 8, 9, 6 }

↳ $N-2$ length $\Rightarrow \underline{7}$.

↳ 2 and 7
↳ missing.

1 to 9 or N

$$\left. \begin{array}{l} a+b=9 \\ [ab]=14 \end{array} \right\}$$

$$(a+b)^2 - 4ab = (a-b)^2$$

without extra space $\rightarrow \checkmark$

{ 3, 4, 1, 5, 8, 9, 6 }

int z = 0;

for (int i = 1; i <= n; i++) {

step 2

step 1

$$z = z \wedge i;$$

```
for(int i=0; i<arr.length; i++) {
```

$$z = z \wedge arr[i];$$

$$z = 1 \wedge 2 \wedge 3 \wedge 4 \wedge 5 \wedge 6 \wedge 7 \wedge 8 \wedge 9$$

$$z = 3 \wedge 4 \wedge 1 \wedge 5 \wedge 8 \wedge 9$$

$$\Rightarrow z = 2 \wedge 7$$

1	→	0001
2	→	0010
3	→	0011
4	→	0100
5	→	0101
6	→	0110
7	→	0111
8	→	1000
9	→	1001

$$\Rightarrow \begin{array}{r} 0010 \\ \wedge 0111 \\ \hline 0101 \end{array}$$

= 5 in dec

③rd step → find right most set bit but we can find any.

$$z = z \& \sim(z-1)$$

$$0101$$

→ 0101 or
iska nega
→ 1011

$3 = 0101$
 $\Rightarrow 11011$
 $3 =$

0	0	0	1	1
---	---	---	---	---

 \rightarrow

4th step.

```

group 1 = 0
for (int i = 1; i <= n; i++) {
    if ((i & 3) == 3) {
        group 1 = group 1 ^ i;
    } else {
        group 2 = group 2 ^ i;
    }
}

```

```

group 2 = 0;
for (int i = 0; i < arr.length; i++) {
    if ((arr[i] & 3) == 3) {
        group 1 = group 1 ^ arr[i];
    } else {
        group 2 = group 2 ^ arr[i];
    }
}

```


}

\ 3

{

}

$$\begin{array}{r}
 1 \wedge 1 \rightarrow \begin{array}{r} 0001 \\ \hline 0001 \end{array} \\
 \text{group 2} \rightarrow \underline{0001}
 \end{array}$$

$$\begin{array}{r}
 2 \wedge 1 \rightarrow \begin{array}{r} 0001 \\ \hline 0010 \end{array} \\
 \underline{0000}
 \end{array}$$

$$\begin{array}{r}
 5 \wedge 1 \rightarrow \begin{array}{r} 0101 \\ \hline 0001 \\ \hline 0001 \end{array}
 \end{array}$$

rightmost bit
= 1
group 1

rightmost bit
set = 0
group 2

$$\rightarrow \cancel{1} \wedge \cancel{3} \wedge \cancel{5} \wedge \cancel{7} \wedge \cancel{9}$$

$$\rightarrow \cancel{2} \wedge \cancel{4} \wedge \cancel{6} \wedge \cancel{8}$$

$$\begin{array}{r}
 3 \wedge 1 \rightarrow \begin{array}{r} 0001 \\ \hline 0011 \end{array} \\
 \rightarrow \underline{0001} \text{ group 1.}
 \end{array}$$

$$\begin{array}{r}
 4 \wedge 1 \rightarrow \begin{array}{r} 0100 \\ \hline 0001 \\ \hline 0000 \end{array}
 \end{array}$$

$$\begin{array}{r}
 6 \rightarrow \begin{array}{r} 0110 \\ \hline 0001 \\ \hline 0000 \end{array}
 \end{array}$$

$$\begin{array}{r}
 7 \rightarrow \begin{array}{r} 01 \\ \hline 00 \\ \hline 00 \end{array}
 \end{array}$$

$$\wedge \cancel{3} \wedge \cancel{5} \wedge \cancel{7} \wedge \cancel{9} \Rightarrow 0$$

$$\wedge \cancel{4} \wedge \cancel{6} \wedge \cancel{8} = 2$$

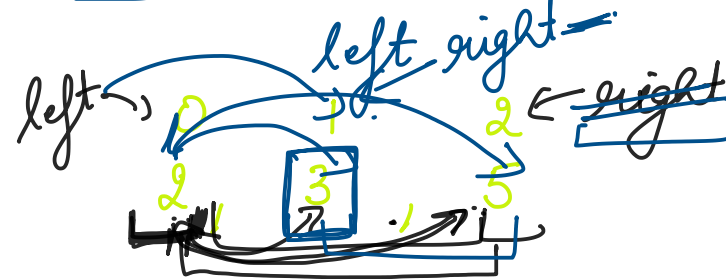


Q \Rightarrow Sorted array.
 arr = { 2, 3, 5 }

Count of distinct rectangles?
 area = 15.

2×2 , 2×3 , ~~3×3~~ , 3×3 , 2×5 .

5×2 , $3 \times 2 \Rightarrow 6$ distinct rectangles.



```
int count = 0;
int area = 15;
while (left <= right) {
```

two pointer approach.

```
    if (arr[left] * arr[right] < area) {
        count = count + 2 * (right - left + 1);
        left++;
    } else {
        right--;
    }
}
```

$$2 \times 5 < 15$$

2×5	2×3
5×2	3×2

$$+ 2 \times 2$$

$$3 \times 3$$