

10칸에 13

→ 6칸 빼칸

↳ 헌재기본  
1 0 3

5개를 3개로 뺄 때  
(with 순서)

= 자리를 5개로 칠 때 5인

3자리수를 한 번째로 count

$$f_0(n) = 0$$

$$f_1(n)_2$$

005

014

0 3 4 n

		$f_2(n) = n+1$
023		
032		
041	$f_3(n)$	
050		
104	$= \sum_{i=0}^n f_2(i)$	
113		
122	$= \sum_{i=0}^n (i+1)$	
131		
140	$= \frac{(n+1)(n+2)}{2}$	
2		

	0	1	2	3	23 index
0	1	1	1	1	$\boxed{0 \rightarrow 3}$
1	0	1	2	3	$\frac{27}{2}$
2	0	1	3	6	$\square$
3	0	1	4	10	$\square$
4	0	1	5	15	
5	0	1	6	21	

6	0	1	7	29
7	0	1	3	36
0	0	1	7	45
0				

224

$$f_3(0) - f_3(3-2)$$

$$+ f_2(6) - f_3(6-2)$$

$$+ f_1(4) - f_1(4-4)$$

a b c

$$\rightarrow (f_3(a+b+c) - f_3(b+c))$$

$$+ (f_2(b+c) - f_2(c))$$

~~19~~ = index //

par sing!

0 0 3

0 1 7

0 2 6

0 3 7

0 4 9 ]

0 5 3

0 6 2

0 7 1

0 8 0

1 0 2 ]

1 1 6

1 2 5

1 3 4

1 4 3

1 5 2  
1 6 1

1 7 0

2 0 6

2 1 5

2 2 4

$$\frac{r(r+1)}{2} + C \quad r \geq C$$

	0	1	2	3	4	5	6	7	8
0	1	1	1	1	1	1	-	-	-
1	0	1	2	3	4	5			
2	0	1	3	6	10	15			
3	0	1	4	10	20	35			
4	0	1	5	15	35	80			
5	0	1	6	21	56	136			
6	0	1	7	28	84				

every  $\Delta$  line  
is  $\Delta$  line  
 $\Delta$  line.

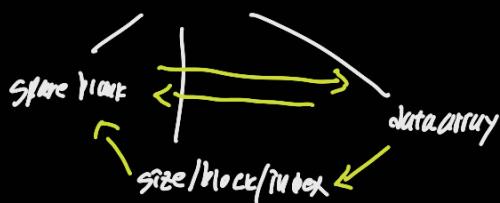
$$f_4(11) = 3/2 \text{ not } 3/11$$

$$100\text{?} \quad 1/10^2 \text{ or } ?$$

$$f_n(x) \rightarrow n \infty \Rightarrow \text{large gap}$$

7	0	1	0	36	120
8	0	1	9	45	165
9	0	1	10	55	220
10	0	1	11	66	286
11	0	1	12	70	364
12	0	1	13	91	
13	0	1	14	105	
14	0	1	15	120	

line data



가로/세로를 관점에서 고려

각 칸의 가능성이 수를 더해 가는 법

Blank/Black 칸은 다 같거. 예상한 대로.

0/1/2/3/4/5/6/7/8/9/10/11/12/13/14/15에 있는 모든 가능한 경우

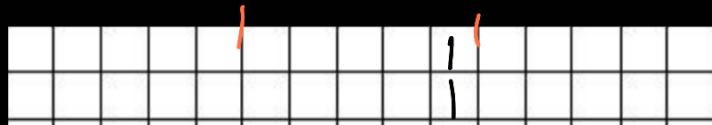
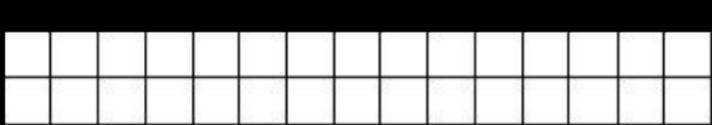
그걸 713827244 정렬

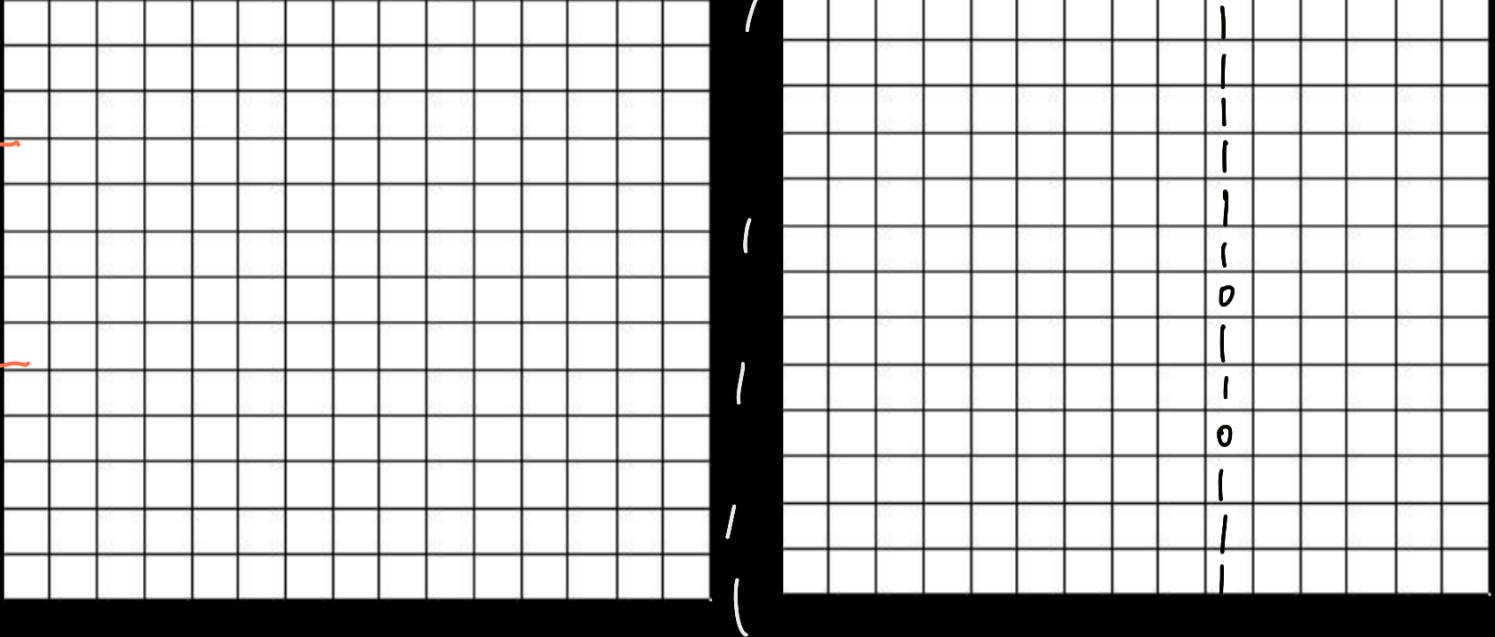
13	12	11	12	11	11	10	120	35	4	1	0	136	3	15	45	15
12	11	10	11	10	9	7	4	1	0	5	1	4	8	14	5	1

13	12		3									0					
11	10		5									0					
36	7		4	3								0					
9	8		7									0					
11	10		5									0					
13	12		3									0					
11	10		5									0					
21	5		1	8								0					
35	4		3	3	3							X					
4	1		7	3	2							0					
10	2		5	4	2							0					
15	4		8	2								X					
6	5		10									0					
55	9		2	3								0					
10	9		6									0					

row map

column map





$f_4(1)$

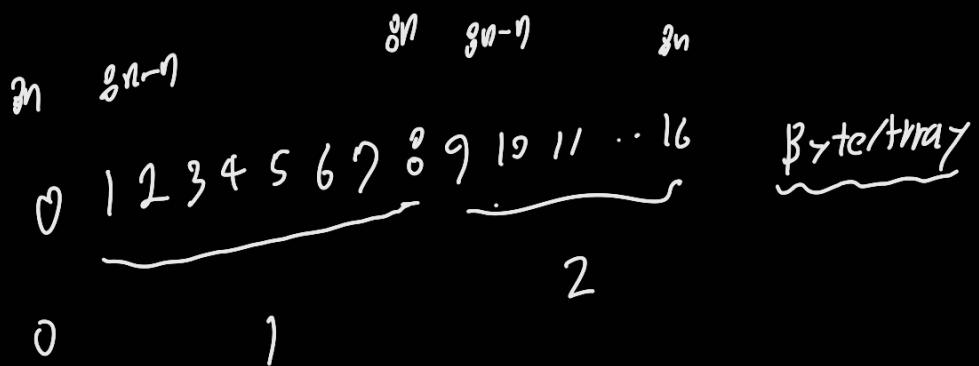
0001      7 3 2  
0010  
0100  
1000      0 0 0 0 0 0 X 0 0 0 X 0 0 X  
0 0 0 0 0 0 X 0 0 0 X X 0 0  
0 0 0 0 0 0 X X 0 0 0 X 0 0  
X 0 0 0 0 0 X 0 0 0 X 0 0  
3 4 4 4 4 4 4 1 2 4 4 2 1 4 3

$15 \rightarrow 7 3 2$      $\overbrace{4,1}^{4}$   
 $\hookrightarrow 11$  2 1 2 1 0 1 1 1 1 1 1 1

5  $\rightarrow$  1 1 3,2 6

b-c	0	1	
blocks	0	1	
0	0	0	0 2
1	1		1 1
			2 0

3 2 0 0 0



$$m \leq k \leq n$$

$$f(k) = n = \left\lfloor \frac{km}{3} \right\rfloor$$

$$n - \frac{7}{3} \leq \frac{k}{3} \leq n$$

$$n \leq \frac{k+n}{8} \leq n + \frac{n}{8}$$

Matrix API package

Index Calculator

L Accumulating Sum,,

LineData

0	1	3	6	10	15
2	4	7	11	16	
3	6	12	19		
9	13	19			
14	19				
					20

- Length / List of Block / index
- Row Data or Block / Blank
- Length / List of Sparse Block  
List of Block

Nonogram Map

L map : Matrix of Block / Blank / Null

↳ indexing  
infinity  
matrix

↳ Could be  
Sequence.

L `lenLine -> linedata`

L `rowMap/columnMap -> represent each frequency`  
`(initial queue)`

L `lineCalcQueue: priority queue.`

제 31 줄 27521 / frequency map을 통한 선별 구현 X

Update 33!

L `possibility set: interset (implemented by Int array and`  
`bitwise shift)`

L `> filtering set process.`

L `replace / multi processing?`

Line indexer

→ array of possibility set

line → row/column map.

`getLine(lineIndex: LineIndexer): linedata`.

`getPossibilitySet(lineIndex: LineIndexer): PossibilitySet`

Line converter (default: length)

rowData ①

spareBlank ②

blockData ③

① → ②

```

data[-1] += 1
if filled data.append(0)
Q->③
①->②->③
②->① / ③->①

```

just make  
 $\textcircled{1} \rightarrow \textcircled{3}$   
 spare block  $\rightarrow$  count index (parsing from query)

$\textcircled{3} \rightarrow \textcircled{2}$   
 $\textcircled{1} \rightarrow \textcircled{2}$  (make from query)

p2 i.e.  $\text{row} \leq \text{column}$

$\text{row} = \text{column}$   $\square$   $\min(\text{row}, \text{column}) = \text{criteria}$

$\text{row} > \text{column}$   $\square$  row-base

$\text{row} < \text{column}$   $\square$  column-base

$\text{criteria} = \min(r, c) + 1$

$\uparrow \text{criteria}$

$\rightarrow \text{basic.getindex}(r-\text{cr}, c)$

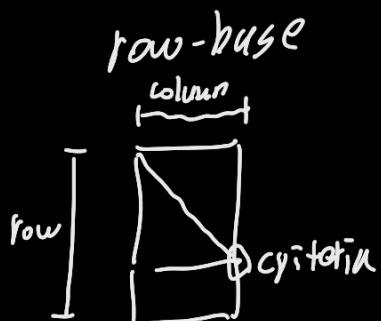
+ sym. count

$\downarrow \text{criteria}$

$\rightarrow \text{basic.getindex}(r, c-\text{cr})$

+ sym. count

else sym. getindex.



0	15	16
12	17	18
3 4 5	19	20
6 7 8 9	21	22
10 11 12 13 14 23 24		
15 16 17 18 19		
20 21 22 23 24		

PH is for take multithreading.

