

## < 분할정복 >

### 1. 증명

$$5 \% 4 = 1$$

$$\underline{10 \% 12 = 2}$$

$$3 \% 34 = 3$$

$$5 \times 5 \% 4 = 1$$

$$\underline{10 \times 10 \% 12 = 4}$$

$$3 \times 3 \% 34 = 9$$

$$5 \times 5 \times 5 \% 4 = 1$$

4

$$3 \times 3 \times 3 \% 34 = 27$$

4

$$3 \times 3 \times 3 \times 3 \% 34 = 13$$

$\left. \begin{array}{l} a < c \\ \Rightarrow b : a \end{array} \right\}$

$$34 \times 2 + 13$$

$$3^5 \% 34 = 5$$

$$34 \times 7 + 5$$

$$l = [10, 10^2 \% 12, 10^2 \% 12, 10^2 \% 12 \dots]$$

$$3^6 \% 34 = 15$$

$$34 \times 21 + 15$$

$$3^7 \% 34 = 11$$

$$34 \times 64 + 11$$

$$3^8 \% 34 = 33$$

$$281$$

$$3^9 \% 34 = 31$$

$$\text{지수법칙: } A^{m+n} = A^m \times A^n$$

$$\text{나머지 분배법칙: } (A \times B) \% C = (A \% C) \times (B \% C) \% C$$

$$10^n \% 12$$

$$= (10^5 \% 12 \times 10^5 \% 12 \times 10) \% 12$$

$$= (10^2 \% 12 \times 10^2 \% 12 \times 10) \% 12 \times (10^2 \% 12 \times 10^2 \% 12 \times 10) \% 12$$

```

1 a, b, c = map(int, input().split())
2
3 def multi(a, b, c):
4     if b == 1:
5         return a%c
6     else:
7         next = multi(a,b//2, c)
8         if b % 2 == 0:
9             return (next*next) % c
10    else:
11        return (next*next*a) % c
12
13 print(multi(a,b,c))

```

$$\textcircled{1} \text{ } \text{multi}(10, 11, 12)$$

$$\textcircled{2} \text{ } ((\text{multi}(10, 5, 2) \times \text{multi}(10, 5, 2) \times 10) \% 12$$

$$\textcircled{3} \text{ } (((\text{multi}(10, 2, 2) \times \text{multi}(10, 2, 2) \times 10) \% 12$$

$$\times (\text{multi}(10, 2, 2) \times \text{multi}(10, 2, 2) \times 10) \% 12 \times 10) \% 12$$

$$\textcircled{4} \text{ } \{(([\text{multi}(10, 1, 2) \times \text{multi}(10, 1, 2)]) \% 12$$

$$\times ([\text{multi}(10, 1, 2) \times \text{multi}(10, 1, 2)]) \% 12) \times 10\} \% 12$$

$$\left[ \{([\text{multi}(10, 1, 2) \times \text{multi}(10, 1, 2)]) \% 12 \right. \\ \left. \times ([\text{multi}(10, 1, 2) \times \text{multi}(10, 1, 2)]) \% 12\} \times 10 \} \% 12 \right] \times 10$$

$$\% 12$$

## 2. 이항계수

$$C_N^k = N! / (k!(N-k)!) \approx 1,000,000,007$$

페르마의 소정리

$p$ 가  $a$ 의 배수가 아니다  
( $= a \neq p$ )

$a$ 가 정수이고  $p$ 가 소수이고  $a+p$  일 때  $a^p \equiv a \pmod{p}$  이며  $a \neq 0$  일 때  $a^{p-1} \equiv 1 \pmod{p}$  이다.

ex)  $3^5 \% 5 = 243 \% 5 = 3$       a를 p로 나눈 나머지  
 $3 \% 5 = 3$

$\Rightarrow p$ 가 소수이고  $a$ 가  $p$ 의 배수가 아니면  $a^{p-1} \equiv 1 \pmod{p}$  이다.

즉,  $a^{p-1}$  을  $p$ 로 나눈 나머지는 1이다.

( $1,000,000,007 \rightarrow$  소수)

여기서  $a$ 를 나누면  
 $a^{p-2} \equiv a^{-1} \pmod{p}$  ( $a \neq 0$ )  
 $\frac{1}{a}(x)$   $a \times \boxed{x} = 1$  여기

$$C_N^k \% p = \frac{N!}{(N-k)!k!} \% p = N!((N-k)!k!)^{-1} \% p = N!((N-k)!k!)^{p-2} \% p$$

$$(N-k+1)!k!^{-1} \% p \quad (N-k+1)!k^{p-2} \% p$$

```

1 def power(a, b):
2     if b == 0:
3         return 1
4     if b % 2:
5         return (power(a, b // 2) ** 2 * a) % p
6     else:
7         return (power(a, b // 2) ** 2) % p
8
9 p = 1000000007
10 N, K = map(int, input().split())
11
12 fact = [1 for _ in range(N + 1)]
13
14 for i in range(2, N + 1):
15     fact[i] = fact[i - 1] * i % p
16
17 A = fact[N]
18 B = (fact[N - K] * fact[K]) % p
19
20 print((A % p) * (power(B, p - 2) % p) % p)

```

$a^b$  계산  
ex)  $3^5 = (3^2)^2 \times 3 = ((3^1)^2)^2 \times 3 = ((3^1) \times 3)^{2 \times 2} \times 3$

5,2

$[1, 1, 1, 1, 1]$        $5! = \frac{5!}{3!2!}$

$[1, 1, 1 \times 2, 1 \times 2 \times 3, 1 \times 2 \times 3 \times 4, 1 \times 2 \times 3 \times 4 \times 5]$

$A = 5! (1 \times 2 \times 3 \times 4 \times 5)$

$B = 3! \times 2!$

$2 \times \square = 1 \quad \square = \frac{1}{2}$  (곱셈에 대한 예선)

$a \div 2 = a \times \frac{1}{2}$

$Z/n: 0, 1, 2, \dots, n-1$

$Z/5: 0, 1, 2, 3, 4$

$\rightarrow$  예선  $n \times (n^{p-2}) \equiv 1$

### 3. 행렬 곱셈

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \\ 5 & 6 \end{bmatrix} \times \begin{bmatrix} -1 & -2 & 0 \\ 0 & 0 & 3 \end{bmatrix} = \begin{bmatrix} -1 & -2 & 6 \\ -3 & -6 & 12 \\ -5 & -10 & 18 \end{bmatrix}$$

### 5. 피보나치 수 6

$$\begin{pmatrix} F_{n+2} \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix} \begin{pmatrix} F_n \\ F_{n+1} \end{pmatrix} \leftarrow \begin{pmatrix} F_{n+2} \\ F_{n+1} \end{pmatrix} = \begin{pmatrix} F_{n+1} + F_n \\ F_{n+1} + 0 \end{pmatrix}$$

$\therefore \begin{pmatrix} F_{n+1} & F_n \\ F_n & F_{n-1} \end{pmatrix} = \begin{pmatrix} 1 & 1 \\ 1 & 0 \end{pmatrix}^n$

```

1 import sys
2 input = sys.stdin.readline
3
4 n = int(input())
5 p = 1000000007
6
7 # n번 째 피보나치 수는 행렬  $(1 \ 1, 1 \ 0)^n$  의 1행 2열 값이다(단, n>=1일때)
8
9 def mul(A, B):
10    n = len(A)
11    Z = [[0]*n for _ in range(n)]
12
13    for row in range(n):
14        for col in range(n):
15            e = 0
16            for i in range(n):
17                e += A[row][i] * B[i][col]
18            Z[row][col] = e % p
19
20    return Z
21
22 def square(A, k):
23    if k == 1:
24        for x in range(len(A)):
25            for y in range(len(A)):
26                A[x][y] %= p
27    return A
28
29 tmp = square(A, k//2)
30 if k % 2:
31     return mul(mul(tmp, tmp), A)
32 else:
33     return mul(tmp, tmp)
34
35 fib_matrix = [[1, 1], [1, 0]]
36 print(square(fib_matrix, n)[0][1])

```

$n=5$

$$Z = [[0], [0], [0], [0], [0]]$$

$$\text{square}\left(\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, 5\right)$$

$$\Rightarrow \text{tmp} = \text{square}\left(\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, 2\right)$$

$$\text{square}\left(\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, 1\right)$$

$$= \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

$$n/2 = 1$$

$$\text{mul}\left(\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}, \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}\right)$$

$$\Rightarrow n=2$$

$$Z = [[0, 0], [0, 1]]$$

$$i=0 \quad 2 \ 1 \quad 1 \ 1$$

$$\begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix} \quad \begin{bmatrix} 1 & 1 \\ 1 & 0 \end{bmatrix}$$

기동계급수행

$$\text{ex) } A=2, k=5 \quad (2^5)$$

$$5/2 = 1$$

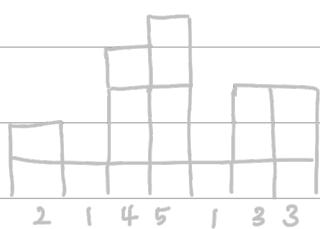
$$\therefore \text{return } 2 \times 2^2 \times 2^2$$

$$\begin{matrix} 2 \times 2 \\ 2 \times 2 \end{matrix}$$

## 6. 히스토그램에서 가장 큰 직사각형

세그먼트 트리 : 여러개의 데이터가 연속적으로 존재할 때 특정 데이터의 합을 구하는 방법

ex)



ex) 최솟값 찾기 : 2 1 4 5 1 3 3



```

67 while(True):
68     temp = list(map(int, input().split()))
69     if temp[0] == 0:
70         break
71     n = temp[0] → [2 1 4 5 1 3 3]
72     arr = temp[1:] → arr(2 1 4 5 1 3 3)
73     tree = [0]*(pow(2,math.ceil(math.log(len(arr),2))+1)-1)
74     # 편하게 n*4를 해도 되지만, 최적 길이로 생성하는 방법은 위와 같다
75
76     init(arr,tree,1,0,len(arr)-1)
77
78     print(dac(0,len(arr)-1))
    
```

$\text{pow}(2, \underline{n}) : 2^n$  반환

→ 밑의 2진 표기

$\text{pow}(2, \text{ceil}(\log_2 7)+1)$

$$= \text{pow}(2, 3) = 2^3 = 8$$

$$\therefore \text{tree} = [0]*8 = [0, 0, 0, 0, \dots, 0]$$

arr = [2 1 4 5 1 3 3]

총 7개

init (arr, tree, 1, 0, 6)

mid = 3

init ( , , 2, 0, 3)

init ( , , 4, 0, 1)

init ( , , 6, 0, 0)  $\Rightarrow \text{tree}[7] = 0$

init ( , , 9, 1, 1)  $\Rightarrow \text{tree}[8] = 1$

init ( , , 5, 2, 3)  $\Rightarrow \text{tree}[5] = 3$

init ( , , 3, 4, 6)

```

1 import math, sys
2 input = sys.stdin.readline
3
4 # 아래의 설정을 해주지 않으면 최대 재귀 제한에 걸려 런타임 오류가 생긴다.
5 sys.setrecursionlimit(10**6)
6
7 # 세그먼트 트리 생성
8 def init(arr,tree,node,start,end):
9     if start == end:
10         tree[node-1] = start → arr에 원소가 1개일 경우
11     else:
12         mid = (start+end)//2
13         init(arr,tree,node*2,start,mid)
14         init(arr,tree,node*2+1,mid+1,end)
15         # node-1은 부모, node*2-1과 node*2는 각각 왼쪽, 오른쪽 자식노드를 의미
16
17         if arr[tree[node-1]] < arr[tree[node*2]]:
18             tree[node-1] = tree[node*2-1]
19         else:
20             tree[node-1] = tree[node*2]
21             # 부모 노드에 더 작은 자식 노드를 입력
22
23 # 구간 최솟값을 찾아주는 쿼리 함수
24 def query(arr,tree,node,start,end,x,y):
25     # 주어진 범위가 전체 범위를 벗어난 경우
26     if x > end or y < start:
27         return -1
28     # 주어진 범위가 전체 범위 안에 포함되는 경우
29     if start >= x and end <= y:
30         return tree[node-1]
31
32     mid = (start+end)//2
33     left = query(arr,tree,node*2,start,mid,x,y)
34     right = query(arr,tree,node*2+1,mid+1,end,x,y)
    
```

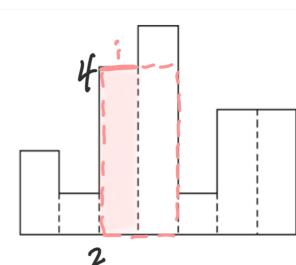
```

36 # 한쪽 노드가 범위를 벗어난 경우 자연스럽게 반대쪽 노드가 선택됨
37     if left == -1:
38         return right
39     elif right == -1:
40         return left
41     else:
42         # 더 작은 값의 인덱스 선택
43         if arr[left] >= arr[right]:
44             return right
45         else:
46             return left
47
48 def dac(start,end):
49     # 해당 구간 범위의 최솟값 인덱스를 구함
50     index = query(arr,tree,1,0,len(arr)-1,start,end)
51
52     # 단일 블럭인 경우
53     if abs(end-start)==0:
54         return arr[index]
55
56     # v1 = 가장 낮은 블럭의 높이 * 히스토그램의 길이
57     v1,v2,v3 = arr[index] * (end-start+1),0,0
58
59     # 범위를 벗어나지 않는 경우 분할
60     if index-1 >= start:
61         v2 = dac(start,index-1)
62     if index+1 <= end:
63         v3 = dac(index+1,end)
64
65     return max(v1,v2,v3)
66

```

## 히스토그램에서 가장 큰 직사각형

풀이 : 직사각형의 높이 고정시키기



⇨ 를 직사각형의 윗면으로 포함하는 가장 큰 직사각형?

→ 해당 막대를 양옆으로 늘릴 때

현재 블록의 왼, 오른쪽의 높이가 h[i] 보다 작은 첫블록까지

stack에 h[i] 높이에서 왼쪽으로 최대한 갈수 있는 idx 저장 (idx, height)

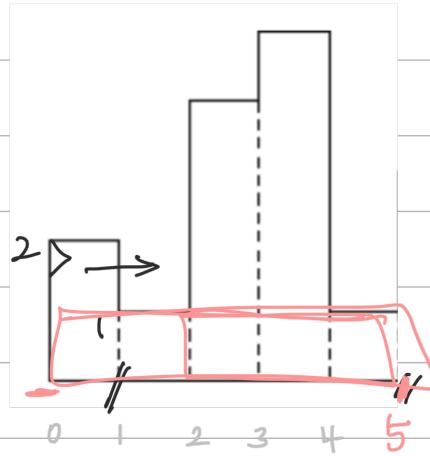
다음 높이가 이전 높이보다 작으면 오른쪽으로 더 갈수 X

→ pop 하는 히스토그램 높이의 직사각형 넓이 구하기

```

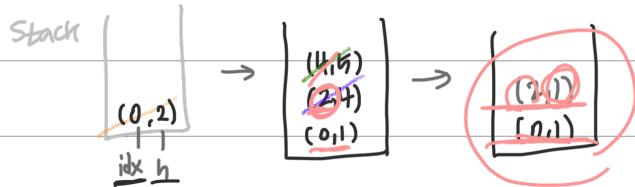
1 import sys
2
3 st = []
4 while True:
5     info = list(map(int, sys.stdin.readline().split()))
6     if info[0] == 0:
7         break
8
9     n = info[0]
10    h = info[1:] 높이
11    temp = [] 스택
12    for i in range(n):
13        cut = i
14        while st and st[-1][1] > h[i]:
15            idx, height = st.pop()
16            temp.append(height * (i - idx))
17            cut = idx
18        st.append((cut, h[i]))
19
20    while st:
21        idx, height = st.pop()
22        temp.append(height * (n - idx))
23
24 print(max(temp))

```



(index 0 1 2 3 4)  
높이 [2 1 4 5 1])

1)  $i=0$  : 스택이 비어있으므로  $(0, 2)$  push



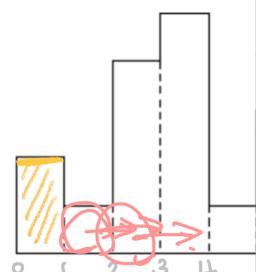
2)  $i=1$  :  $2 > h[1]$   $\Rightarrow (0, 2)$  pop

$$idx=0, height=2 \Rightarrow 높이 : 2 \times (1-0)=2$$

스택에  $(0, 1)$  push

높이 1이 직사각형이 왼쪽으로 갖는다는 점

3)  $i=2$  :  $h[2] > 1$   $\Rightarrow (2, 4)$  push



4)  $i=3$  :  $h[3] > 1$   $\Rightarrow (3, 5)$  push

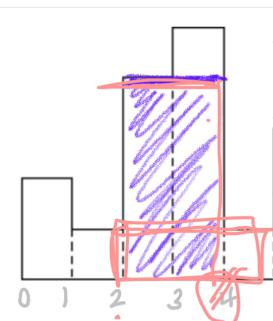
5)  $i=4$  :  $5 > h[4]$   $\Rightarrow (3, 5)$  pop

$$idx=3, height=5 \Rightarrow 높이 = 5 \times (4-3) = 5$$

$4 > h[3] \Rightarrow (2, 4)$  pop

$$idx=2, height=4 \Rightarrow 높이 = 4 \times (4-2) = 8$$

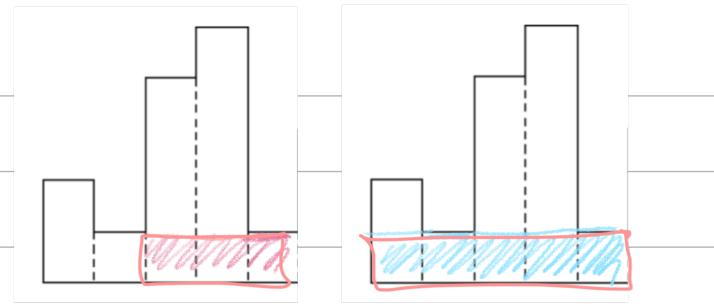
스택에  $(2, 1)$  push



스택에 넣은 값 처리

1) (2,1) pop  $\Rightarrow$   $\text{넓이} = 1 \times (5-2) = 3$

2) (0,1) pop  $\Rightarrow$   $\text{넓이} = 1 \times (5-0) = 5$



$\text{temp} = [2, 5, 8, 3, 5]$