Revision.

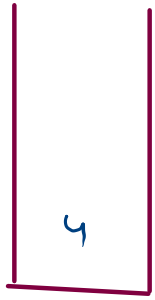
```
5   public static int [] ngel(int [] A){
6       int [] ans = new int[A.length];
7       ans[0] = -1;
8       Stack<Integer> st = new Stack<>();
9       st.push(0);     //I am inserting index not value
10
11      for(int i = 1; i < ans.length; i++){
12          while(st.size() != 0 && A[st.peek()] <= A[i]){
13              st.pop();
14          }
15          if(st.size() == 0){
16              ans[i] = -1;
17          }
18          else {
19              ans[i] = A[st.peek()];
20          }
21
22          st.push(i);
23      }
24
25
26      return ans;
27  }
```

expected -1 5 3 3 -1 6

5 3 1 2 6 4

0 1 2 3 4 5

-1	5	3	3	-1	6



A[4] ≤ A[5]

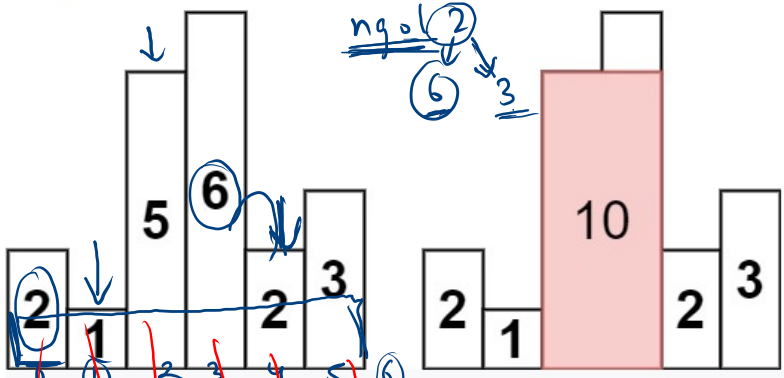
6 ≤ 4 ✗

84. Largest Rectangle in Histogram

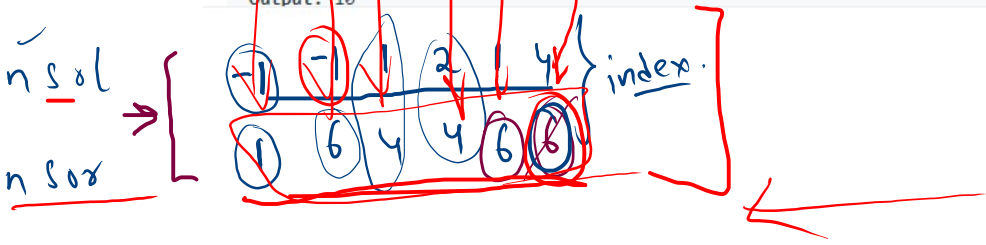
Hard 15036 215 Add to List Share

Given an array of integers `heights` representing the histogram's bar height where the width of each bar is 1, return *the area of the largest rectangle in the histogram.*

Example 1:




```
Input: heights = [2,1,5,6,2,3]
Output: 10
```



nsol

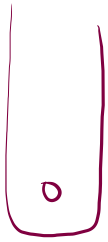
nsor

(ngol)

$A = h \times w$ ↑ maximise.

ngol

2	1	5	6	2	3
0	1	2	3	4	5
-1	1				



ngol.

A[peek] ≤ A[i]

pop.

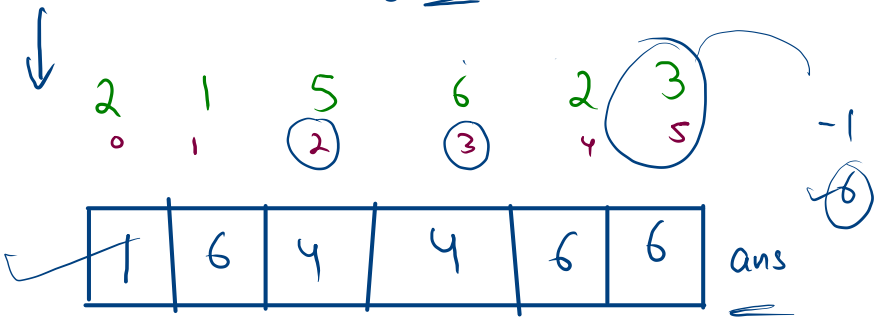
next smaller on right.

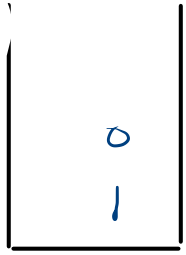
ngor

n = 6

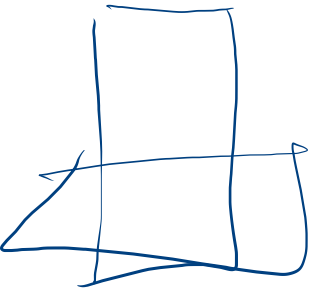
```java
public int [] nsor(int [] A){
    int n = A.length;
    int [] ans = new int[n];
    ans[n-1] = n;
    Stack<Integer> st = new Stack<>();
    st.push(n-1);      //I am inserting index not value

    for(int i = n-2; i >= 0; i--){
        while(st.size() != 0 && A[st.peek()] > A[i]){
            st.pop();
        }
        if(st.size() == 0){
            ans[i] = n;
        }
        else {
            ans[i] = st.peek();
        }

        st.push(i);
    }
    return ans;
}
```

ngol

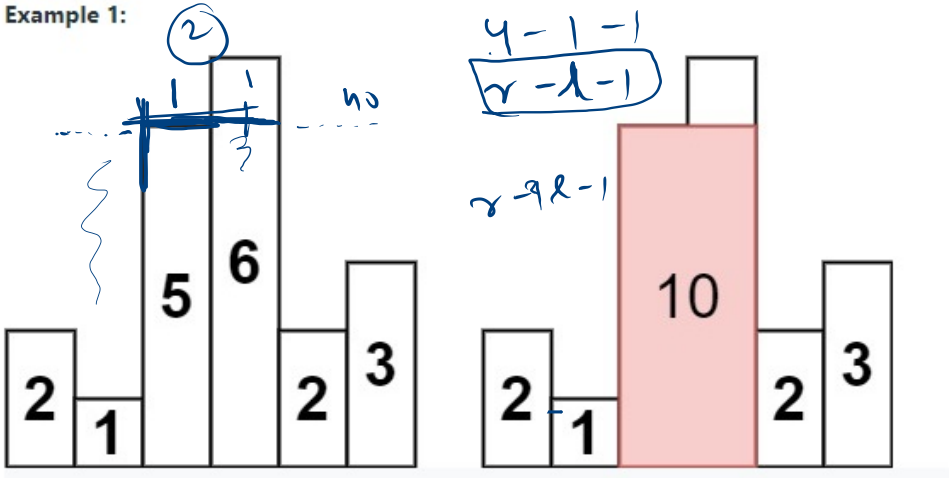




2 1 5 6 2 3



Example 1:



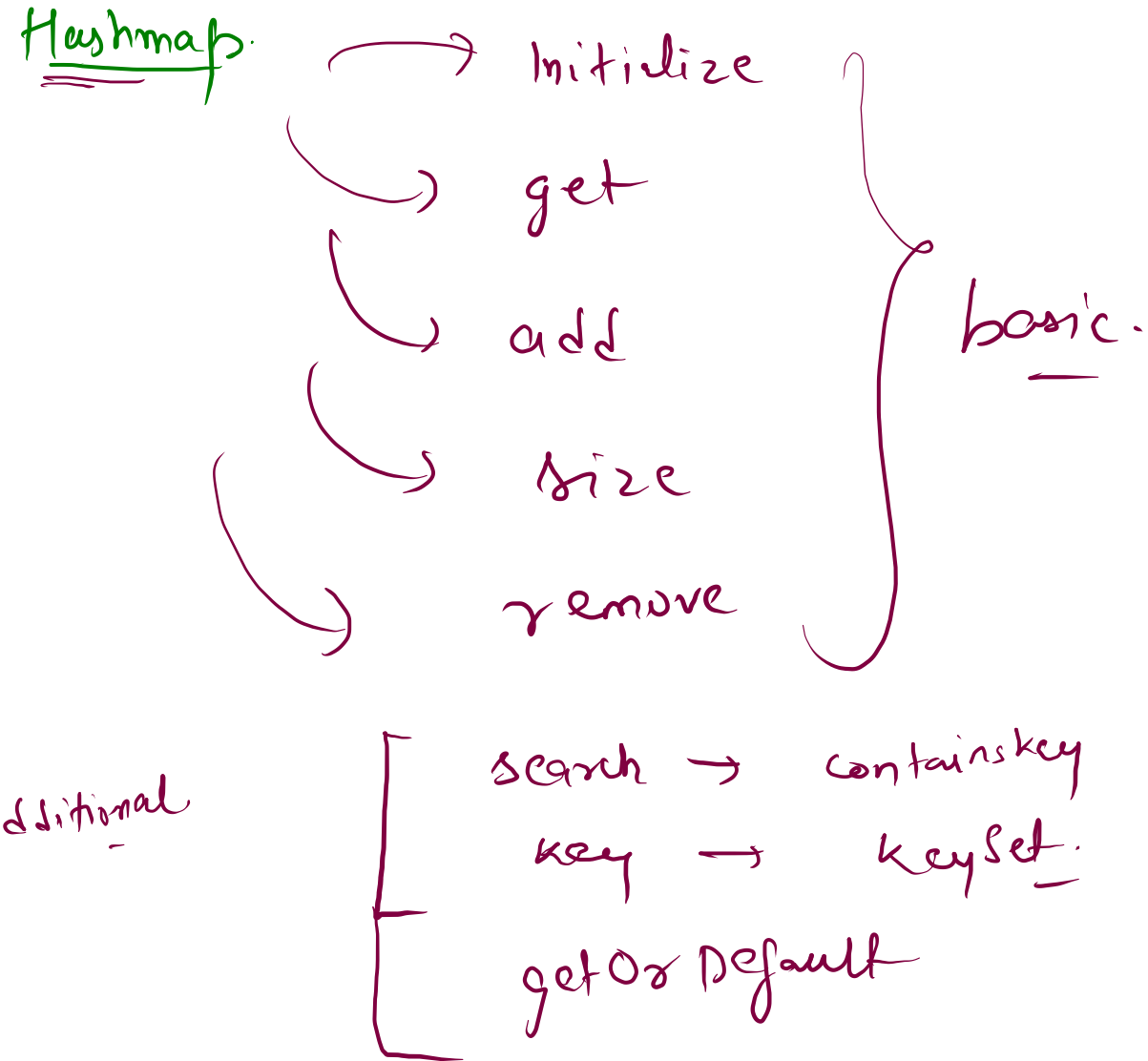

```
Input: heights = [2,1,5,6,2,3]
Output: 10
```

Hash map

Class.

Java. → part of collection.

List → Array List


Hashmap.
Initialize
get
add
size
remove
basic.
additional
search → containskey
key → keyset.
getOrDefault

Hashmap

<k, v> pair

population of City.

✓	✓
"Delhi"	24000
"BOM"	16000
"Kolkata"	12000
"Patna"	23000

≈ O(1)

O(constant)

↓

O(1)

Array.

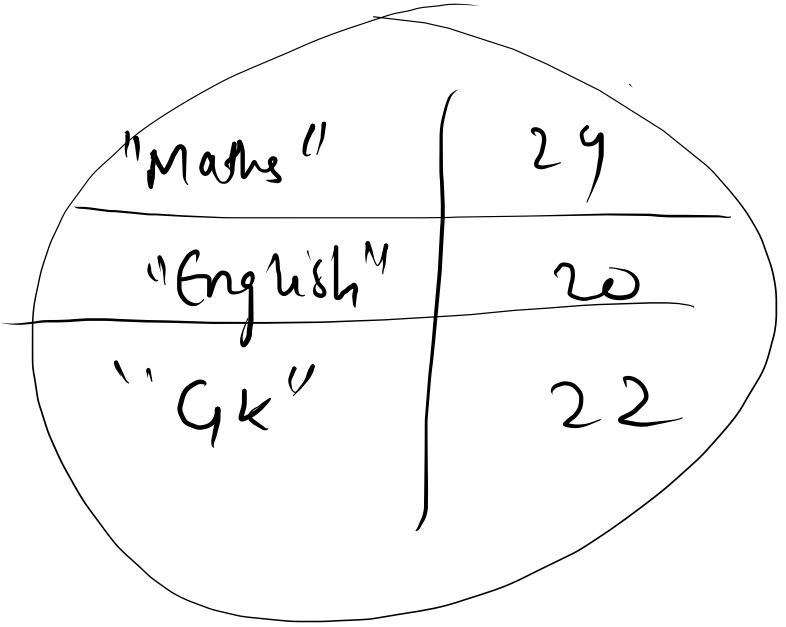
A[2]

→ O(1)

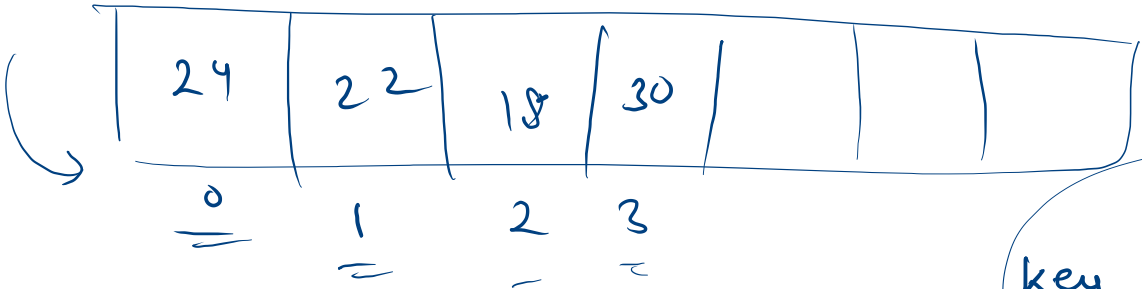
Delhi

→ O(1)

Marks.



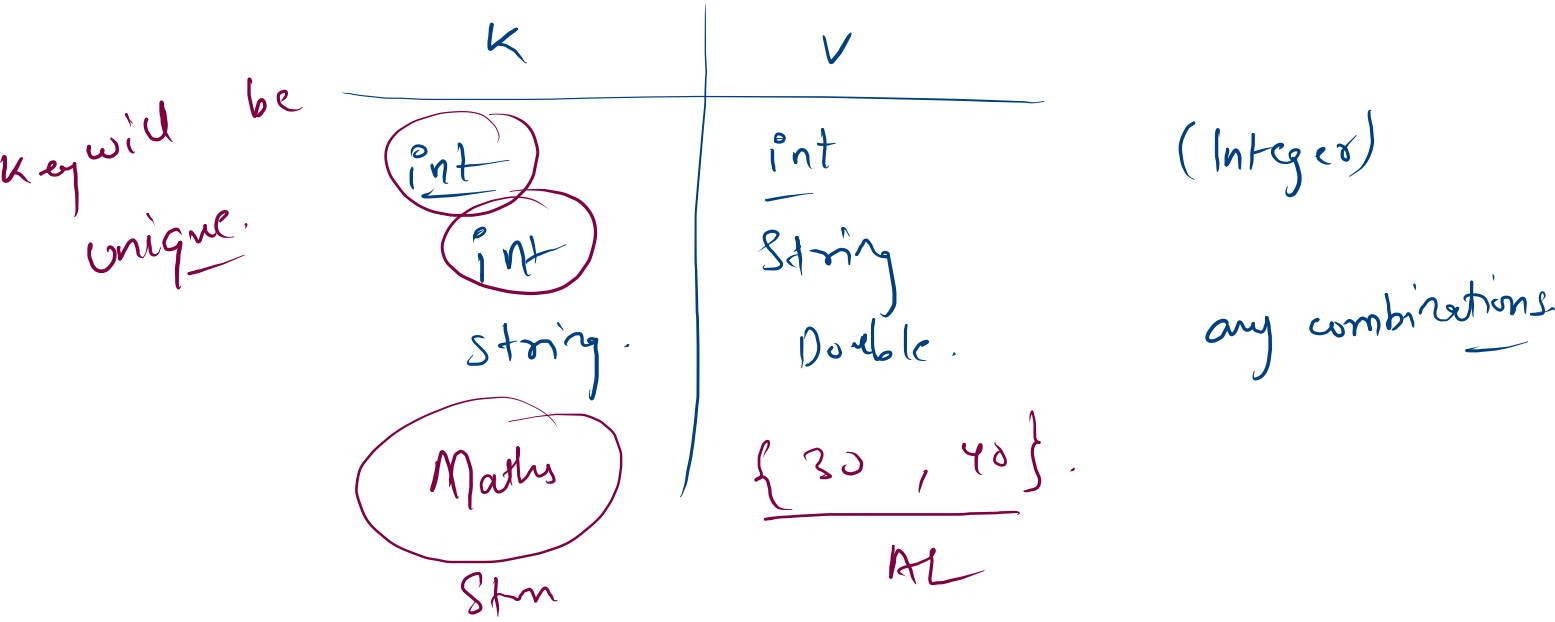
Marks:



key will act as index.

Hashmap:

Aman	24
Praveen	18
Sagar	30


K
V
keywid be unique.
int
int
string.
Maths
Stm
int
String
Double.
{30, 40}.
AL
(Integer)
any combinations

put → insert (new)

→ update (already).

```java
public class Main
{
    public static void main(String[] args) {
        //inti: Population of city
//      ArrayList<String> arr = new ArrayList<>();

        HashMap<String, Integer> hm = new HashMap<>();

        //add value: put
        hm.put("Delhi", 112);
        hm.put("Mumbai", 283);
        hm.put("Patna", 579);
        hm.put("Kolkata", 556);
        hm.put("Delhi", 432);

        //get
        System.out.println(hm.get("Patna"));

        //size
        System.out.println("Size is : "  + hm.size());

        //remove
        hm.remove("Mumbai");
    }
}
```

```java
        HashMap<String, Integer> hm = new HashMap<>();

        //add value: put
        hm.put("Delhi", 112);
        hm.put("Mumbai", 283);
        hm.put("Patna", 579);
        hm.put("Kolkata", 556);
        hm.put("Delhi", 432);

        //containsKey

        System.out.println(hm.containsKey("CITY"));

        //fetch all the keys
        for(String key : hm.keySet()){
            System.out.println(key);
        }

        //getOrDefault

        int val = hm.getOrDefault("CITY", 100);

        System.out.println("Result is " + val);
```

Word Meaning

You are required to create a **dictionary** consisting of word and its meaning.

Take an integer N as input and **Continue** the process untill **Case 4** is not achieved.

- If N==1, take **word** and **meaning** as input from user and **add** it to the dictionary.
- If N==2, take a **word** as input from the user and print its **meaning**, if the word is not found print **-1**.
- If N==3, take a **word** as input from the user and delete it from the dictionary.
- If N==4, **Close** the dictionary(**Exit** the program).

Sample Input 0

```
1
Geekster
Coding
1
Geek
Coder
2
Geek
3
Geekster
2
Geekster
4
```

Sample Output 0

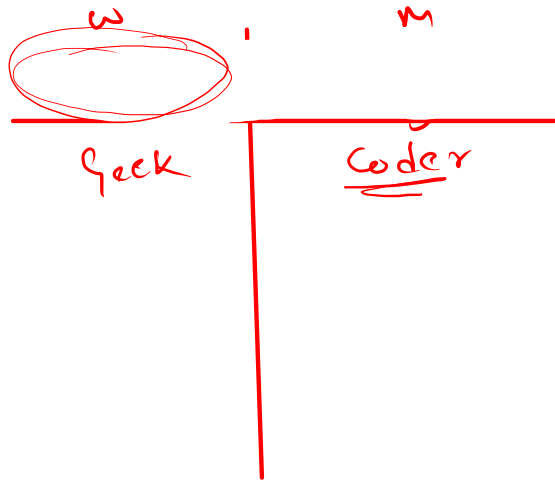
```
Coder
-1
```

N→1 put

N→2 get / -1

N→3 remove.

N→4 exit / Return



```java
import java.io.*;
import java.util.*;

public class Solution {

    public static void main(String[] args) {
       Scanner scn = new Scanner(System.in);

        HashMap<String , String > hm = new HashMap<>();


        while(true){
            int n = scn.nextInt();

            if(n == 1){
                //put
                String word = scn.next();
                String meaning  = scn.next();

                hm.put(word, meaning);


            }
            else if(n == 2){
                String word = scn.next();
                System.out.println(hm.getOrDefault(word, "-1"));
            }
            else if(n == 3){
                String word = scn.next();
                hm.remove(word);
            }
            else if(n == 4){
                break;
            }
        }
    }
}
```

Same Number Same Frequency

Sample Input 0

```
10
4 5 -3 8 -3 4 4 -3 6 4
```

Sample Output 0

```
-3
4
```

4 5 -3 8 -3 4 4 -3 6 4

k	v
4	4
5	1
-3	3
8	1
6	1

abs(k) = val

4 → 4

-3 → 3

abs value of key should be equal to value

print key.

1. freq map.