

---

---

---

---

---



STL  $\rightarrow$  Binary Search

always  
Sorted  
array

$\hookrightarrow$  lower bound

$\hookrightarrow$  upper bound

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

lower-bound ( v.begin(), v.end(), 2 )

$\downarrow$   
Index  $\rightarrow$  2

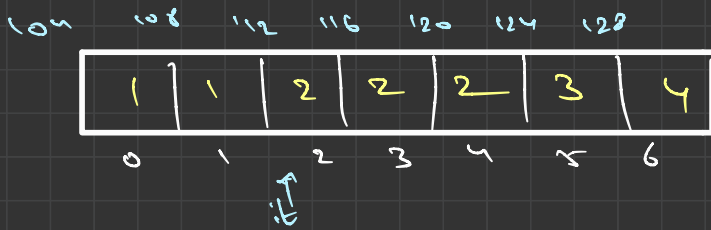
what to  
search

upper-bound ( v.begin(), v.end(), 2 )

$\uparrow$   
Index  $\rightarrow$  5

what to  
search

$\hookrightarrow$  last occurrence  $\hookleftarrow$  next pointer



auto it = lower\_bound(v.begin(), v.end(), 2)

↑  
we don't know  
what they return  
then we use auto

cout << \*it ;   → to get address

for print index

cout << "index : " << it - v.begin() ;  
  return index

it - v.begin()

⇓

112 - 104 = 8 bytes

1 int ⇒ 4 byte

8/4 = 2 index