

Contents

1	Section1	1
1.1	Binary search	1
2	Section2	1
2.1	thm	1

1 Section1

1.1 Binary search

```

1 // c++ code
2 #include <bits/stdc++.h>
3 lower_bound(a, a + n, k);    //最左邊 ≥ k 的位置
4 upper_bound(a, a + n, k);    //最左邊 > k 的位置
5 upper_bound(a, a + n, k) - 1; //最右邊 ≤ k 的位置
6 lower_bound(a, a + n, k) - 1; //最右邊 < k 的位置
7 [lower_bound, upper_bound) //等於 k 的範圍
8 equal_range(a, a+n, k);
9
10 // 從小到大
11 priority_queue<int, vector<int>, greater<int>>>pq
12
13 insert(it,x)//向vector的任意迭代器it處插入一個元素x，時間複雜度O(N)
14 erase(it)//刪除迭代器為it處的元素，erase(first,last)刪除一個區間[first,last)內的所有元素，時間複雜度均為O(N)
15
16 set
17 insert(x) //將x插入set中 O(log(n))
18 count(x) //回傳x是否存在於set中() O(log(n))
19 erase(x) //刪除在set中的x O(log(n))
20 clear() //刪除set中所有元素 O(n)
21 empty() //回傳是否為空 O(1)
22 size() //回傳共有幾個元素 O(1)
23
24 map
25 insert(x) //將x這個pair插入map中 O(log(n))
26 count(x) //回傳x這個key是否在map中 O(log(n))
27 erase(x) //刪除在map中key為x的 O(log(n))
28
29
30 #include <bits/stdc++.h>
31 using namespace std;
32
33 int main(){
34     set<int>s;
35     for(int i = 0; i < 10; i++){
36         s.insert(i);
37     }
38     cout << "lower bound: " << *s.lower_bound(5) <<
39         '\n';// 5
40     cout << "upper bound: " << *s.upper_bound(5) <<
41         '\n';// 6
42
43     if(s.lower_bound(20) == s.end()){
44         cout << "all elements are less than 20\n";
45     }
46 }
```

2 Section2

2.1 thm

• 中測試

$$\bullet \sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$$