

Contents

1	basic	1
1.1	binarySearch	1
1.2	stringstream	1
2	Section1	1
2.1	basic	1
3	Section2	1
3.1	thm	1

1 basic

1.1 binarySearch

```

1 #include <bits/stdc++.h>
2 using namespace std;
3
4 int binary_search(const vector<int> &data, int key)
5 {
6     int low = 0;
7     int high = data.size()-1;
8     while (low <= high)
9     {
10         int mid = int((low + high) / 2);
11         if (key == data[mid])
12             return mid;
13         else if (key > data[mid])
14             low = mid + 1;
15         else
16             high = mid - 1;
17     }
18     return -1;
19 }
20
21 int main()
22 {
23     vector<int> data = {1, 9, 2, 7, 4, 10, 3, 8, 5,
24                        6};
25     int key = 7;
26
27     sort(data.begin(), data.end());
28
29     int ret = binary_search(data, key);
30     if (ret == -1)
31         cout << "找不到\n";
32     else
33         cout << "找到索引值" << ret << "\n";
34     //lower_bound(a, a + n, k);    //最左邊 ≥ k 的位置
35     //upper_bound(a, a + n, k);    //最左邊 > k 的位置
36     //upper_bound(a, a + n, k) - 1; //最右邊 ≤ k 的位置
37     //lower_bound(a, a + n, k) - 1; //最右邊 < k 的位置
38     //[lower_bound, upper_bound) //等於 k 的範圍
39     //equal_range(a, a+n, k);
40 }
```

1.2 stringstream

```

1 #include <sstream>
2 using namespace std;
3
4 int main()
5 {
6     stringstream ss;
7     int num = 1234;
8     string output;
9
10    ss << num;
11    ss >> output; //integer to string
12
13    string_to_int << ss;
```

```

14    string_to_int >> num; //string to integer
15
16    ss.str("");
17    ss.clear(); //initialization
18
19    return 0;
20 }
```

2 Section1

2.1 basic

```

1 // c++ code
2 #include <bits/stdc++.h>
3 using namespace std;
4
5 int main() {
6     // test comment
7     cout << "test string\n";
8 }
```

3 Section2

3.1 thm

- 中文測試
- $\sum_{i=1}^n i^2 = \frac{n(n+1)(2n+1)}{6}$