

## Contents

1 Libraries	1
1.1 c libraries	1
1.2 algorithm	1
1.3 map	1
1.4 set	1
1.5 vector	1
1.6 string	1
2 Algorithms	1
2.1 最短路	1
2.1.1 Bellman-Ford	1
2.1.2 Dijkstra's	1
2.2 LIS - Longest Increasing Subsequence	1
3 Formula	1
3.1 thm	1

```

35 */
36
37 bool prev_permutation (BidirectionalIterator
    first, BidirectionalIterator last);
38 /* Rearranges the elements in the range
    [first,last) into the previous
    lexicographically-ordered permutation, then
    returns
    * true if could rearrange as a
    * lexicographically smaller permutation
    * false if arrangement is the largest
    * possible (and sorted in descending order)
    */
39
40
41
42
43
44 return 0;
45 }

```

## 1 Libraries

### 1.1 c libraries

### 1.2 algorithm

```

1 #include <algorithm>
2 using namespace std;
3
4 int main() {
5
6     void sort (RandomAccessIterator first,
7               RandomAccessIterator last);
8     /* Sorts the elements in the range [first,last)
9        into ascending order
10        * In N*lg(N) complexity
11        */
12
13     ForwardIterator lower_bound (ForwardIterator
14                                   first, ForwardIterator last, const T& val);
15     /* Returns an iterator pointing to the first
16        element in the range [first,last) which is >=
17        val
18        * In lg(N)+1 complexity
19        * requires sorted elements
20        */
21
22     ForwardIterator upper_bound (ForwardIterator
23                                   first, ForwardIterator last, const T& val);
24     /* Returns an iterator pointing to the first
25        element in the range [first,last) which is >
26        val
27        * In lg(N)+1 complexity
28        * requires sorted elements
29        */
30
31     pair<ForwardIterator, ForwardIterator> equal_range
32         (ForwardIterator first, ForwardIterator last,
33          const T& val);
34     /* Returns the bounds of the subrange with all
35        the elements == val of the range [first,last)
36        * return type equivalent to pair<
37        * lower_bound(), upper_bound>
38        * In 2*lg(N)+1 complexity
39        * requires sorted elements
40        */
41
42     bool next_permutation (BidirectionalIterator
43                             first, BidirectionalIterator last);
44     /* Rearranges the elements in the range
45        [first,last) into the next lexicographically
46        greater permutation, then returns
47        * true if could rearrange as a
48        * lexicographically greater permutation
49        * false if no greater arrangement than the
50        * previous (and sorted in ascending order)
51        * In N/2 complexity

```

### 1.3 map

### 1.4 set

### 1.5 vector

### 1.6 string

## 2 Algorithms

### 2.1 最短路

#### 2.1.1 Bellman-Ford

#### 2.1.2 Dijkstra's

### 2.2 LIS - Longest Increasing Subsequence

## 3 Formula

### 3.1 thm

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