

Contents

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

1 Libraries

1.1 cstdlib

```

1 #include <cstdlib>
2 using namespace std;
3
4 int main() {
5
6     // Functions
7     // String conversion
8     double atof (const char* str); //
9     // Convert string to double; return 0.0
10    // if no conversion
11    int atoi (const char * str); //
12    // Convert string to integer;
13    long int atol ( const char * str ); //
14    // Convert string to long integer;
15    // return 0 if no conversion
16    long long int atoll ( const char * str );
17    // Convert string to long long
18    // integer; return 0 if no conversion
19    double strtod (const char* str, char**
20    endptr); // Convert string to double;
21    // return 0.0 if no conversion,
22    // HUGE_VAL(cmath) if out of range
23    float strttof (const char* str, char**
24    endptr); // Convert string to float
25
26    strtol
27    Convert string to long integer (function)
28
29    strtold
30    Convert string to long double (function)
31
32    strtoll
33    Convert string to long long integer (function)
34
35    strtoul
36    Convert string to unsigned long integer (function)
37
38    strtoull
39    Convert string to unsigned long long integer
40    (function)
41
42    Pseudo-random sequence generation
43
44    rand
45    Generate random number (function)
46
47    srand
48    Initialize random number generator (function)
49
50    Searching and sorting

```

```

42 bsearch
43     Binary search in array (function)
44
45 qsort
46     Sort elements of array (function)
47
48 Integer arithmetics
49
50 abs
51     Absolute value (function)
52
53 div
54     Integral division (function)
55
56 labs
57     Absolute value (function)
58
59 ldiv
60     Integral division (function)
61
62 llabs
63     Absolute value (function)
64
65 lldiv
66     Integral division (function)
67
68 Macro constants
69
70 NULL
71     Null pointer (macro)
72
73 RAND_MAX
74     Maximum value returned by rand (macro)
75
76 Types
77
78 div_t
79     Structure returned by div (type)
80
81 ldiv_t
82     Structure returned by ldiv (type)
83
84 lldiv_t
85     Structure returned by lldiv (type)
86
87 size_t
88     Unsigned integral type (type)
89
90 return 0;
91 }

```

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

```

```

13  * In lg(N)+1 complexity
14  *   requires sorted elements
15  */
16
17  ForwardIterator upper_bound (ForwardIterator
    first, ForwardIterator last, const T& val);
18  /* Returns an iterator pointing to the first
    element in the range [first,last) which is >
    val
19  * In lg(N)+1 complexity
20  *   requires sorted elements
21  */
22
23  pair<ForwardIterator,ForwardIterator> equal_range
    (ForwardIterator first, ForwardIterator last,
    const T& val);
24  /* Returns the bounds of the subrange with all
    the elements == val of the range [first,last)
25  *   return type equivalent to pair<
    lower_bound(), upper_bound>
26  * In 2*lg(N)+1 complexity
27  *   requires sorted elements
28  */
29
30  bool next_permutation (BidirectionalIterator
    first, BidirectionalIterator last);
31  /* Rearranges the elements in the range
    [first,last) into the next lexicographically
    greater permutation, then returns
32  *   true if could rearrange as a
    lexicographically greater permutation
33  *   false if no greater arrangement than the
    previous (and sorted in ascending order)
34  * In N/2 complexity
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
39  *   true if could rearrange as a
    lexicographically smaller permutation
40  *   false if arrangement is the largest
    possible (and sorted in descending order)
41  * In N/2 complexity
42  */
43
44  return 0;
45 }

```

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

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