

## Contents

### 1 Shell Script

#### 2 Libraries

- 2.1 cstdlib . . . . .
- 2.2 algorithm . . . . .
- 2.3 map . . . . .
- 2.4 set . . . . .
- 2.5 vector . . . . .
- 2.6 string . . . . .

#### 3 Algorithms

- 3.1 最短路 . . . . .
  - 3.1.1 Bellman-Ford . . . . .
  - 3.1.2 Dijkstra's . . . . .
- 3.2 LIS - Longest Increasing Subsequence . . . . .

#### 4 Formula

- 4.1 thm . . . . .

## 1 Shell Script

```
1 #!/bin/bash
2 clear
3 g++ $1.cpp -DDBG -o $1
4 if [[ "$?" == "0" ]]; then
5     echo Running
6     ./$1 <$1.in> $1.out
7     echo END
8 fi
```

## 2 Libraries

### 2.1 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 strtof (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)
```

```
27 strtoull
28     Convert string to unsigned long long integer
29     (function)
30
31 Pseudo-random sequence generation
32
33 rand
34     Generate random number (function)
35
36 srand
37     Initialize random number generator (function)
38
39 Searching and sorting
40
41 bsearch
42     Binary search in array (function)
43
44 qsort
45     Sort elements of array (function)
46
47 Integer arithmetics
48
49 abs
50     Absolute value (function)
51
52 div
53     Integral division (function)
54
55 labs
56     Absolute value (function)
57
58 ldiv
59     Integral division (function)
60
61 llabs
62     Absolute value (function)
63
64 lldiv
65     Integral division (function)
66
67 Macro constants
68
69 NULL
70     Null pointer (macro)
71
72 RAND_MAX
73     Maximum value returned by rand (macro)
74
75 Types
76
77 div_t
78     Structure returned by div (type)
79
80 ldiv_t
81     Structure returned by ldiv (type)
82
83 lldiv_t
84     Structure returned by lldiv (type)
85
86 size_t
87     Unsigned integral type (type)
88
89 return 0;
90
91 }
```

### 2.2 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
52        */
53
54     bool prev_permutation (BidirectionalIterator
55                             first, BidirectionalIterator last);
56     /* Rearranges the elements in the range
57        [first,last) into the previous
58        lexicographically-ordered permutation, then
59        returns
60        * true if could rearrange as a
61        lexicographically smaller permutation
62        * false if arrangement is the largest
63        possible (and sorted in descending order)
64        * In N/2 complexity
65        */
66
67     return 0;
68 }

```

## 2.3 map

## 2.4 set

## 2.5 vector

## 2.6 string

# 3 Algorithms

## 3.1 最短路

### 3.1.1 Bellman-Ford

### 3.1.2 Dijkstra's

## 3.2 LIS - Longest Increasing Subsequence

# 4 Formula

## 4.1 thm

• 中文測試

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