

Contents

1 Todo	
2 Contest Setup	
2.1 vimrc	1
2.2 bashrc	1
2.3 C++ template	2
2.4 Java template	2
3 Reminder	
4 Useful code	
4.1 Fast Exponentiation	3
4.2 GCD	3
4.3 Extended Euclidean Algorithm	3
4.4 STL quick reference	3
4.4.1 Map / Set	3
4.4.2 String	3
5 Search	
5.1 Binary Search	3
5.1.1 Find key	3
5.1.2 Upper / lower Bound	3
5.2 折半完全列舉	3
5.3 Two-pointer 爬行法	3
6 Basic data structure	
6.1 1D BIT	3
6.2 2D BIT	3
6.3 Union Find	3
6.4 Segment Tree	3
7 Dynamic Programming	
8 Tree	
8.1 LCA	3
9 Graph	
9.1 Articulation point / edge	3
9.2 BCC vertex	3
9.3 BCC edge	3
9.4 SCC	3
9.5 Shortest Path	3
9.5.1 Dijkstra	3
9.5.2 SPFA	3
9.5.3 Bellman-Ford	3
9.6 Flow	3
9.6.1 Max Flow (Dinic)	3
9.6.2 Min-Cut	3
9.6.3 Min Cost Max Flow	3
9.6.4 Maximum Bipartite Graph	3
10 String	
10.1 KMP	3
10.2 Z Algorithm	3
10.3 Trie	3
10.4 Suffix Array	3
11 Geometry	
11.1 Template	3
11.1.1 Point / Line	3
11.1.2 Intersection	3
11.2 Half-plane intersection	3
11.3 Convex Hull	3

1 Todo

1. Add code and complexity
2. Add brief explanations

2 Contest Setup

2.1 vimrc

```

3 1 set number           " Show line numbers
3 2 set mouse=a         " Enable inaction via mouse
3 3 set showmatch        " Highlight matching brace
3 4 set cursorline      " Show underline
3 5 set cursorcolumn    " Show vertical column
3 6 highlight
3 7 set ruler           " Show row and column ruler
3 8 information
3 9 filetype on "enable file detection
3 10 syntax on "syntax highlight
3 11
3 12 set autoindent      " Auto-indent new lines
3 13 set shiftwidth=4    " Number of auto-indent
3 14 spaces
3 15 set smartindent    " Enable smart-indent
3 16 set smarttab       " Enable smart-tabs
3 17 set softtabstop=4  " Number of spaces per Tab
3 18
3 19 " -----Optional-----
3 20 set undolevels=10000 " Number of undo levels
3 21 set scrolloff=5      "auto scroll on the bottom 5
3 22 lines
3 23
3 24 set hlsearch         " Highlight all search
3 25 results
3 26 set smartcase       " Enable smart-case search
3 27 set ignorecase      " Always case-insensitive
3 28 set incsearch       " Searches for strings
3 29 incrementally
3 30
3 31 highlight Comment ctermfg=cyan
3 32 set showmode
3 33
3 34 set encoding=utf-8
3 35 set fileencoding=utf-8
3 36 scriptencoding=utf-8

```

contest_setup/vimrc

2.2 bashrc

```

1 alias g++="g++ -Wall -Wextra -O2"

```

contest_setup/bashrc

2.3 C++ template

```

1 #include <bits/stdc++.h>
2 //LLONG_MIN LLONG_MAX INT_MIN INT_MAX
3
4 #ifdef _WIN32
5 #define lld "I64d"
6 #else
7 #define lld "lld"
8 #endif
9
10 using namespace std;
11
12 #define x first
13 #define y second
14
15 typedef long long int ll;
16 typedef pair<int, int> ii;
17
18 int main()
19 {
20     return 0;
21 }

```

contest_setup/main.cpp

2.4 Java template

```

1 import java.io.*;
2 import java.util.*;
3
4 public class Main
5 {
6     public static void main(String[] args)
7     {
8         MyScanner sc = new MyScanner();
9         out = new PrintWriter(new
10             BufferedOutputStream(System.out));
11         // Start writing your solution here.
12
13         // Stop writing your solution here.
14         out.close();
15     }
16
17     public static PrintWriter out;
18
19     public static class MyScanner
20     {
21         BufferedReader br;
22         StringTokenizer st;
23
24         public MyScanner()
25         {
26             br = new BufferedReader(new
27                 InputStreamReader(System.in));
28         }
29
30         boolean hasNext()
31         {
32             while (st == null || !st.
33                 hasMoreElements()) {
34                 try {
35                     st = new StringTokenizer(
36                         br.readLine());
37                 } catch (Exception e) {
38                     return false;
39                 }
40             }
41             return true;
42         }
43
44         String next()
45         {
46             if (hasNext())
47                 return st.nextToken();
48         }
49     }
50 }

```

```

44         return null;
45     }
46
47     int nextInt()
48     {
49         return Integer.parseInt(next());
50     }
51
52     long nextLong()
53     {
54         return Long.parseLong(next());
55     }
56
57     double nextDouble()
58     {
59         return Double.parseDouble(next());
60     }
61
62     String nextLine()
63     {
64         String str = "";
65         try {
66             str = br.readLine();
67         } catch (IOException e) {
68             e.printStackTrace();
69         }
70         return str;
71     }
72 }
73 }

```

contest_setup/Main.java

3 Reminder

1. Read the problem statements carefully. Input and output specifications are crucial!
2. Estimate the **time complexity** and **memory complexity** carefully.
3. Time penalty is 20 minutes per WA, **don't rush!**
4. Sample test cases must all be tested and passed before every submission!
5. Test the corner cases, such as 0, 1, -1. Test all edge cases of the input specification.

4 Useful code

4.1 Fast Exponentiation

4.2 GCD

小心負數!

4.3 Extended Euclidean Algorithm

4.4 STL quick reference

4.4.1 Map / Set

4.4.2 String

5 Search

5.1 Binary Search

5.1.1 Find key

5.1.2 Upper / lower Bound

5.2 折半完全列舉

5.3 Two-pointer 爬行法

6 Basic data structure

6.1 1D BIT

6.2 2D BIT

6.3 Union Find

6.4 Segment Tree

Hehe

7 Dynamic Programming

8 Tree

8.1 LCA

9 Graph

9.1 Articulation point / edge

9.2 BCC vertex

9.3 BCC edge

9.4 SCC

9.5 Shortest Path

9.5.1 Dijkstra

9.5.2 SPFA

9.5.3 Bellman-Ford

9.6 Flow

9.6.1 Max Flow (Dinic)

9.6.2 Min-Cut

9.6.3 Min Cost Max Flow

9.6.4 Maximum Bipartite Graph

10 String

10.1 KMP

10.2 Z Algorithm

10.3 Trie

10.4 Suffix Array

11 Geometry

11.1 Template

11.1.1 Point / Line

11.1.2 Intersection

11.2 Half-plane intersection

11.3 Convex Hull