

Contents

1	Todo	
2	Contest Setup	
2.1	vimrc	
2.2	bashrc	
2.3	C++ template	
2.4	Java template	
3	Reminder	
4	Useful code	
4.1	Fast Exponentiation	
4.2	GCD	
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	
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	

1 Todo

1. Add code and complexity
2. Add brief explanations

2 Contest Setup

2.1 vimrc

```

1 set number          " Show line numbers
2 set mouse=a         " Enable inaction via mouse
3 set showmatch       " Highlight matching brace
4 set cursorline      " Show underline
5 set cursorcolumn    " highlight vertical column
6
7 filetype on "enable file detection
8 syntax on "syntax highlight
9
10 set autoindent      " Auto-indent new lines
11 set shiftwidth=4    " Number of auto-indent spaces
12 set smartindent     " Enable smart-indent
13 set smarttab        " Enable smart-tabs
14 set softtabstop=4   " Number of spaces per Tab
15
16 " -----Optional-----
17
18 set undolevels=10000 " Number of undo levels
19 set scrolloff=5     " Auto scroll
20
21 set hlsearch        " Highlight all search results
22 set smartcase       " Enable smart-case search
23 set ignorecase      " Always case-insensitive
24 set incsearch       " Searches for strings incrementally
25
26 highlight Comment ctermfg=cyan
27 set showmode
28
29 set encoding=utf-8
30 set fileencoding=utf-8
31 set 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
3 using namespace std;
4
5 #define x first
6 #define y second
7
8 typedef long long int ll;
9 typedef pair<int, int> ii;
10
11 1111111111222222222233333333334444444445555555555
12
13 int main()
14 {
15     return 0;
16 }

```

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.hasMoreElements
33                     ()) {
34                     try {
35                         st = new StringTokenizer(br.
36                             readLine());
37                     } catch (Exception e) {
38                         return false;
39                     }
40                     return true;
41                 }
42             }
43
44             String next()
45             {
46                 if (hasNext())
47                     return st.nextToken();
48                 return null;
49             }
50
51             int nextInt()
52             {
53                 return Integer.parseInt(next());
54             }
55
56             long nextLong()
57             {
58                 return Long.parseLong(next());
59             }
60
61             double nextDouble()
62             {
63                 return Double.parseDouble(next());
64             }
65
66             String nextLine()
67             {
68                 String str = "";
69                 try {
70                     str = br.readLine();
71                 } catch (IOException e) {
72                     e.printStackTrace();
73                 }
74                 return str;
75             }
76         }
77     }
78 }

```

```

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