

# Contents

<b>1</b>	<b>Todo</b>	<b>1</b>
<b>2</b>	<b>Contest Setup</b>	<b>1</b>
2.1	vimrc	1
2.2	bashrc	1
2.3	C++ template	1
2.4	Java template	1
<b>3</b>	<b>Reminder</b>	<b>2</b>
<b>4</b>	<b>Useful code</b>	<b>2</b>
4.1	Fast Exponentiation	2
4.2	GCD	2
4.3	Extended Euclidean Algorithm	3
4.4	STL quick reference	3
4.4.1	Map / Set	3
4.4.2	String	3
<b>5</b>	<b>Search</b>	<b>3</b>
5.1	Binary Search	3
5.1.1	Find key	3
5.1.2	Upper / lower Bound	3
5.2	折半完全列举	3
5.3	Two-pointer 爬行法	3
<b>6</b>	<b>Basic data structure</b>	<b>3</b>
6.1	1D BIT	3
6.2	2D BIT	3
6.3	Union Find	3
6.4	Segment Tree	3
<b>7</b>	<b>Dynamic Programming</b>	<b>3</b>
<b>8</b>	<b>Tree</b>	<b>3</b>
8.1	LCA	3
<b>9</b>	<b>Graph</b>	<b>3</b>
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
<b>10</b>	<b>String</b>	<b>3</b>
10.1	KMP	3
10.2	Z Algorithm	3
10.3	Trie	3
10.4	Suffix Array	3
<b>11</b>	<b>Geometry</b>	<b>3</b>
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

```

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
12   spaces
13 set smartindent     " Enable smart-indent
14 set smarttab        " Enable smart-tabs
15 set softtabstop=4   " Number of spaces per Tab
16
17 " -----Optional-----
18
19 set undolevels=10000 " Number of undo levels
20 set scrolloff=5     " Auto scroll
21
22 set hlsearch        " Highlight all search results
23 set smartcase       " Enable smart-case search
24 set ignorecase      " Always case-insensitive
25 set incsearch       " Searches for strings
26   incrementally
27
28 highlight Comment ctermfg=cyan
29 set showmode
30
31 set encoding=utf-8
32 set fileencoding=utf-8
33 set scriptencoding=utf-8

```

contest\_setup/vimrc

### 2.2 bashrc

```
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 |             boolean hasNext()
30 |             {
31 |                 while (st == null || !st.
32 |                 hasMoreElements()) {
33 |                     try {
34 |                         st = new StringTokenizer(
35 |                         br.readLine());
36 |                     } catch (Exception e) {
37 |                         return false;
38 |                     }
39 |                 }
40 |                 return true;
41 |             }
42 |
43 |             String next()
44 |             {
45 |                 if (hasNext())
46 |                     return st.nextToken();
47 |                 return null;
48 |             }
49 |
50 |             int nextInt()
51 |             {
52 |                 return Integer.parseInt(next());
53 |             }
54 |
55 |             long nextLong()
56 |             {
57 |                 return Long.parseLong(next());
58 |             }
59 |
60 |             double nextDouble()
61 |             {
62 |                 return Double.parseDouble(next());
63 |             }
64 |
65 |             String nextLine()
66 |             {
67 |                 String str = "";
68 |                 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