Introductory Lecture
Topics: basics,resources,stl,bitwise tricks,gdb

League of Programmers

IIT Kanpur

September 30, 2011

Aim of the Discussion Series

Aim of the Discussion Series

- Discussion camp not a lecture series. You need to show motivation.

Aim of the Discussion Series

- Discussion camp not a lecture series. You need to show motivation.
- To help you code better and faster, introduce you to new tricks mostly to help you do well in programming contests.

Language Issues

- Language We stress on: C++

Language Issues

- Language We stress on: C++
- Slower languages like java and python time out on many judges.

Language Issues

- Language We stress on: C++
- Slower languages like java and python time out on many judges.
- C has too restrictive and does not support stl/ templates / classes.

Popular Websites to practice on

Popular Websites to practice on

- Compete against Indian coders in live contests Codechef

Popular Websites to practice on

- Compete against Indian coders in live contests Codechef
- As problem archive Codeforces/Spoj/UVA/Topcoder/Project Euler

Popular Websites to practice on

- Compete against Indian coders in live contests Codechef
- As problem archive Codeforces/Spoj/UVA/Topcoder/Project Euler
- Short Programming Contests Codeforces/Topcoder

Popular Websites to practice on

- Compete against Indian coders in live contests Codechef
- As problem archive Codeforces/Spoj/UVA/Topcoder/Project Euler
- Short Programming Contests Codeforces/Topcoder
- Tougher Problems acm.sgu.ru/Codechef Monthly challenge/IOI/ACM ICPC previous years archive/Topcoder level 3

Popular Contests to participate in

- India: ACM ICPC/ IOPC/ Shastra OPC/ Codecraft/ Codechef Monthly Challenge/ Codechef Cook off

Popular Contests to participate in

- India: ACM ICPC/ IOPC/ Shastra OPC/ Codecraft/ Codechef Monthly Challenge/ Codechef Cook off
- World: Topcode SRMs/ Codeforces/ Facebook Hacker Cup/ Google Code Jam

Common Problems

- Overflow

Common Problems

- Overflow Use data type of appropriate size. Choose data type of every variable carefully based on the constraints of the problem
- Passing Multi Dimensional Arrays

Common Problems

- Overflow Use data type of appropriate size. Choose data type of every variable carefully based on the constraints of the problem
- Passing Multi Dimensional Arrays
  Solution: either use vector stl(convenient) or global arrays of large enough size(faster)

Common Problems

- Overflow Use data type of appropriate size. Choose data type of every variable carefully based on the constraints of the problem
- Passing Multi Dimensional Arrays
  Solution: either use vector stl(convenient) or global arrays of large enough size(faster)
- Comparing Doubles

Common Problems

- Overflow Use data type of appropriate size. Choose data type of every variable carefully based on the constraints of the problem
- Passing Multi Dimensional Arrays
  Solution: either use vector stl(convenient) or global arrays of large enough size(faster)
- Comparing Doubles
  Solution: Always keep cushion of a small ϵ.

Common Problems

- Overflow Use data type of appropriate size. Choose data type of every variable carefully based on the constraints of the problem
- Passing Multi Dimensional Arrays
  Solution: either use vector stl(convenient) or global arrays of large enough size(faster)
- Comparing Doubles
  Solution: Always keep cushion of a small ϵ.
- Segmentation faults

Common Problems

- Overflow Use data type of appropriate size. Choose data type of every variable carefully based on the constraints of the problem
- Passing Multi Dimensional Arrays
  Solution: either use vector stl(convenient) or global arrays of large enough size(faster)
- Comparing Doubles
  Solution: Always keep cushion of a small ϵ.
- Segmentation faults
  Solution: Use gdb ⟨*gdb tutorial*⟩

Common Problems

- Overflow Use data type of appropriate size. Choose data type of every variable carefully based on the constraints of the problem
- Passing Multi Dimensional Arrays
  Solution: either use vector stl(convenient) or global arrays of large enough size(faster)
- Comparing Doubles
  Solution: Always keep cushion of a small ϵ.
- Segmentation faults
  Solution: Use gdb ⟨*gdb tutorial*⟩
- From Your side

How to parse a problem???

- Understand what the program is expected to do.

How to parse a problem???

- Understand what the program is expected to do.
- Understand the Input/Output format and use **exactly** that format

How to parse a problem???

- Understand what the program is expected to do.
- Understand the Input/Output format and use **exactly** that format
- Meaning of constraints

How to parse a problem???

- Understand what the program is expected to do.
- Understand the Input/Output format and use **exactly** that format
- Meaning of constraints
- What do time limit and memory limit mean??

How to parse a problem???

- Understand what the program is expected to do.
- Understand the Input/Output format and use **exactly** that format
- Meaning of constraints
- What do time limit and memory limit mean??
- 1 sec $\approx (1-2) * 10^8$ operations

How to parse a problem???

- Understand what the program is expected to do.
- Understand the Input/Output format and use **exactly** that format
- Meaning of constraints
- What do time limit and memory limit mean??
- 1 sec $\approx (1-2) * 10^8$ operations
- x MB $\approx x/4 * 10^6$ sized int arrays

How to parse a problem???

- Understand what the program is expected to do.
- Understand the Input/Output format and use **exactly** that format
- Meaning of constraints
- What do time limit and memory limit mean??
- 1 sec $\approx (1 - 2) * 10^8$ operations
- x MB $\approx x/4 * 10^6$ sized int arrays
  Not all operations are equally fast:

How to parse a problem???

- Understand what the program is expected to do.
- Understand the Input/Output format and use **exactly** that format
- Meaning of constraints
- What do time limit and memory limit mean??
- 1 sec $\approx (1-2)*10^8$ operations
- x MB $\approx x/4*10^6$ sized int arrays
  Not all operations are equally fast:
  operations on unsigned ints/long long are faster

How to parse a problem???

- Understand what the program is expected to do.
- Understand the Input/Output format and use **exactly** that format
- Meaning of constraints
- What do time limit and memory limit mean??
- 1 sec $\approx (1-2) * 10^8$ operations
- x MB $\approx x/4 * 10^6$ sized int arrays
  Not all operations are equally fast:
  operations on unsigned ints/long long are faster
  bitwise operators and shift operators $(\&|\wedge >><<)$

How to parse a problem???

- Understand what the program is expected to do.
- Understand the Input/Output format and use **exactly** that format
- Meaning of constraints
- What do time limit and memory limit mean??
- 1 sec $\approx (1 - 2) * 10^8$ operations
- x MB $\approx x/4 * 10^6$ sized int arrays
  Not all operations are equally fast:
  operations on unsigned ints/long long are faster
  bitwise operators and shift operators $(\&|\wedge >><<)$
  Using too much memory ($>$ 10 MB) slows down programmes

STL

Website: http://www.cplusplus.com/reference

- what is template ?

STL

Website: http://www.cplusplus.com/reference

- what is template ?
- Vector: Dynamic size arrays with insertion/deletion to/from arbitrary locations

STL

Website: http://www.cplusplus.com/reference

- what is template ?
- Vector: Dynamic size arrays with insertion/deletion to/from arbitrary locations
- Set: Just like a mathematical set.
  Problem:spoj.pl/problems/WEIRDFN
  Problem:http://www.spoj.pl/problems/HOMO/
  Problem:spoj.pl/problems/HISTOGRA

STL

Website: http://www.cplusplus.com/reference

- what is template ?
- Vector: Dynamic size arrays with insertion/deletion to/from arbitrary locations
- Set: Just like a mathematical set.
  Problem:spoj.pl/problems/WEIRDFN
  Problem:http://www.spoj.pl/problems/HOMO/
  Problem:spoj.pl/problems/HISTOGRA
- Map: Advanced set
  Problem:spoj.pl/problems/SUBSEQ

STL

Website: http://www.cplusplus.com/reference

- what is template ?
- Vector: Dynamic size arrays with insertion/deletion to/from arbitrary locations
- Set: Just like a mathematical set.
  Problem:spoj.pl/problems/WEIRDFN
  Problem:http://www.spoj.pl/problems/HOMO/
  Problem:spoj.pl/problems/HISTOGRA
- Map: Advanced set
  Problem:spoj.pl/problems/SUBSEQ
- string class and stringstream to parse strings
  useful mostly for topcoder SRM problems

STL

Website: http://www.cplusplus.com/reference

- what is template ?
- Vector: Dynamic size arrays with insertion/deletion to/from arbitrary locations
- Set: Just like a mathematical set.
  Problem:spoj.pl/problems/WEIRDFN
  Problem:http://www.spoj.pl/problems/HOMO/
  Problem:spoj.pl/problems/HISTOGRA
- Map: Advanced set
  Problem:spoj.pl/problems/SUBSEQ
- string class and stringstream to parse strings
  useful mostly for topcoder SRM problems
- memset/fill: initialize arrays

STL

Website: http://www.cplusplus.com/reference

- what is template ?
- Vector: Dynamic size arrays with insertion/deletion to/from arbitrary locations
- Set: Just like a mathematical set.
  Problem:spoj.pl/problems/WEIRDFN
  Problem:http://www.spoj.pl/problems/HOMO/
  Problem:spoj.pl/problems/HISTOGRA
- Map: Advanced set
  Problem:spoj.pl/problems/SUBSEQ
- string class and stringstream to parse strings
  useful mostly for topcoder SRM problems
- memset/fill: initialize arrays
- Queue can be use as queue and stack

STL

Website: http://www.cplusplus.com/reference

- what is template ?
- Vector: Dynamic size arrays with insertion/deletion to/from arbitrary locations
- Set: Just like a mathematical set.
  Problem:spoj.pl/problems/WEIRDFN
  Problem:http://www.spoj.pl/problems/HOMO/
  Problem:spoj.pl/problems/HISTOGRA
- Map: Advanced set
  Problem:spoj.pl/problems/SUBSEQ
- string class and stringstream to parse strings
  useful mostly for topcoder SRM problems
- memset/fill: initialize arrays
- Queue can be use as queue and stack
- bitset(next section)

STL

Website: http://www.cplusplus.com/reference

- what is template ?
- Vector: Dynamic size arrays with insertion/deletion to/from arbitrary locations
- Set: Just like a mathematical set.
  Problem:spoj.pl/problems/WEIRDFN
  Problem:http://www.spoj.pl/problems/HOMO/
  Problem:spoj.pl/problems/HISTOGRA
- Map: Advanced set
  Problem:spoj.pl/problems/SUBSEQ
- string class and stringstream to parse strings
  useful mostly for topcoder SRM problems
- memset/fill: initialize arrays
- Queue can be use as queue and stack
- bitset(next section)

Introduction to bitwise operators

Introduction to bitwise operators

- Numbers are stored in binary and processing on bits is way faster

Introduction to bitwise operators

- Numbers are stored in binary and processing on bits is way faster
- Our weapons:
  $<<$ $(left\ shift)$, $>>$ $(right\ shift)$, $\&(bitwise\ and)$, $|(bitwise\ or)$, $\hat{}(bitwise\ xor)$ $\tilde{}(bitwise\ not)$

Introduction to bitwise operators

- Numbers are stored in binary and processing on bits is way faster
- Our weapons:
  $<<$ (*left shift*), $>>$ (*right shift*), &(*bitwise and*), |(*bitwise or*), ^(*bitwise xor*) ~(*bitwise not*)
- speed up the code by upto 100 times.

Introduction to bitwise operators

- Numbers are stored in binary and processing on bits is way faster
- Our weapons:
  $<<$ (*left shift*), $>>$ (*right shift*), &(*bitwise and*), $|$(*bitwise or*), ^(*bitwise xor*) ~(*bitwise not*)
- speed up the code by upto 100 times. caution: try to use bitwise operations on unsigned integers only

Beauty of Bitwise

Beauty of Bitwise

- Example:
  Any subset of $\{0,1\ldots31\}$ is a single int
  Do set union/intersection/complement in one operation
  increment /decrement all elements by x in one operation

Beauty of Bitwise

- Example:
  Any subset of $\{0,1 \ldots 31\}$ is a single int
  Do set union/intersection/complement in one operation
  increment /decrement all elements by x in one operation
- Even more:
  Find if $x \in S$.
  Generate all subsets of S in $2^{|S|}$ time
  Generate all subsets of $\{1..n\}$ changing one bit at a time
  Generate all subsets of S which have exactly t elements
  Count the number of elements of elements in a set S
  Remove smallest element from S
  check if $|S| = 1$

Beauty of Bitwise

- Example:
  Any subset of $\{0,1\ldots31\}$ is a single int
  Do set union/intersection/complement in one operation
  increment /decrement all elements by x in one operation
- Even more:
  Find if $x \in S$.
  Generate all subsets of S in $2^{|S|}$ time
  Generate all subsets of $\{1..n\}$ changing one bit at a time
  Generate all subsets of S which have exactly t elements
  Count the number of elements of elements in a set S
  Remove smallest element from S
  check if $|S| = 1$
- **Never** multiply or divide or take remainder modulo power of 2

Practice Probems

More tricks: graphics.stanford.edu/~seander/bithacks.html
lab.polygonal.de/2007/05/10/bitwise-gems-fast-integer-math
bits.stephan-brumme.com
Problems:
codechef.com/problems/TEAMSEL
spoj.pl/problems/PIZZALOC
acm.sgu.ru/problem.php?contest=0&problem=249
codechef.com/problems/SEQUENCE
spoj.pl/problems/NGM2
spoj.pl/problems/VILLAGES
bitwise dp:
community.topcoder.com/stat?c=problem_statement&pm=6725&rd=101
community.topcoder.com/stat?c=problem_statement&pm=6095&rd=991
community.topcoder.com/stat?c=problem_statement&pm=6400&rd=100