Essentials of Competitive Coding

Suyash Agrawal

September 19, 2017

What is it?

Here is what wikipedia says:

What is it?

Here is what wikipedia says:

> *Competitive programming is a mind sport usually held over the Internet or a local network, involving participants trying to program according to provided specifications.*

What is it?

Here is what wikipedia says:

> *Competitive programming is a mind sport usually held over the Internet or a local network, involving participants trying to program according to provided specifications.*

Why is it useful?

What is it?

Here is what wikipedia says:

> *Competitive programming is a mind sport usually held over the Internet or a local network, involving participants trying to program according to provided specifications.*

Why is it useful?

- You will become a better, more efficient, and less error-prone coder.

What is it?

Here is what wikipedia says:

> *Competitive programming is a mind sport usually held over the Internet or a local network, involving participants trying to program according to provided specifications.*

Why is it useful?

- You will become a better, more efficient, and less error-prone coder.
- Your technical programming interviews will be a piece of cake.

What is it?

Here is what wikipedia says:

> *Competitive programming is a mind sport usually held over the Internet or a local network, involving participants trying to program according to provided specifications.*

Why is it useful?

- You will become a better, more efficient, and less error-prone coder.
- Your technical programming interviews will be a piece of cake.
- You win prizes and trips to places around the world as you compete with and meet other motivated and smart students.

Pre-requisites

Here are some pre requisites:

Pre-requisites

Here are some pre requisites:

- Language basic of either C++ or Java.

Pre-requisites

Here are some pre requisites:

- Language basic of either C++ or Java.
- Basic algorithmic skills.

Pre-requisites

Here are some pre requisites:

- Language basic of either C++ or Java.
- Basic algorithmic skills.

And most importantly,

Pre-requisites

Here are some pre requisites:

- Language basic of either C++ or Java.
- Basic algorithmic skills.

  And most importantly,

- Perseverance

Mathematics

Some knowledge of mathematics is required too, like:

Mathematics

Some knowledge of mathematics is required too, like:

- Modulo Arithmetic *Basic Number Theory*

Mathematics

Some knowledge of mathematics is required too, like:

- Modulo Arithmetic *Basic Number Theory*
- Fast Exponentiation

Mathematics

Some knowledge of mathematics is required too, like:

- Modulo Arithmetic *Basic Number Theory*
- Fast Exponentiation
- Complexity Analysis

Mathematics

Some knowledge of mathematics is required too, like:

- Modulo Arithmetic *Basic Number Theory*
- Fast Exponentiation
- Complexity Analysis
- Sieve of Eratosthenes

Mathematics

Some knowledge of mathematics is required too, like:

- Modulo Arithmetic *Basic Number Theory*
- Fast Exponentiation
- Complexity Analysis
- Sieve of Eratosthenes
- Basic Geometry

Mathematics

Some knowledge of mathematics is required too, like:

- Modulo Arithmetic *Basic Number Theory*
- Fast Exponentiation
- Complexity Analysis
- Sieve of Eratosthenes
- Basic Geometry

Don't worry even if you have no idea about these. You will easily pick these up as you practice along.

Problem Structure

Basic structure of the problem is as follows:

Problem Structure

Basic structure of the problem is as follows:

- **Problem Statement**: Describes the problem and what output is to be generated. Usually in form of a long story from which you have to extract the essence.

Problem Structure

Basic structure of the problem is as follows:

- **Problem Statement**: Describes the problem and what output is to be generated. Usually in form of a long story from which you have to extract the essence.
- **Input**: Describes the format of input. Read carefully as missing out any minor detail lands you in wrong answer zone.

Problem Structure

Basic structure of the problem is as follows:

- **Problem Statement**: Describes the problem and what output is to be generated. Usually in form of a long story from which you have to extract the essence.
- **Input**: Describes the format of input. Read carefully as missing out any minor detail lands you in wrong answer zone.
- **Output**: Describes the output format of the problem . Just like above, this one also should be read carefully.

Problem Structure

Basic structure of the problem is as follows:

- **Problem Statement**: Describes the problem and what output is to be generated. Usually in form of a long story from which you have to extract the essence.
- **Input**: Describes the format of input. Read carefully as missing out any minor detail lands you in wrong answer zone.
- **Output**: Describes the output format of the problem . Just like above, this one also should be read carefully.
- **Contraints**: These can include constraints on input, time, memory, code size, etc.

Problem Structure

Basic structure of the problem is as follows:

- **Problem Statement**: Describes the problem and what output is to be generated. Usually in form of a long story from which you have to extract the essence.
- **Input**: Describes the format of input. Read carefully as missing out any minor detail lands you in wrong answer zone.
- **Output**: Describes the output format of the problem . Just like above, this one also should be read carefully.
- **Contraints**: These can include constraints on input, time, memory, code size, etc.
- **Time Limit**: See if your algorithm can work in this range.

Problem Structure

Basic structure of the problem is as follows:

- **Problem Statement**: Describes the problem and what output is to be generated. Usually in form of a long story from which you have to extract the essence.
- **Input**: Describes the format of input. Read carefully as missing out any minor detail lands you in wrong answer zone.
- **Output**: Describes the output format of the problem . Just like above, this one also should be read carefully.
- **Contraints**: These can include constraints on input, time, memory, code size, etc.
- **Time Limit**: See if your algorithm can work in this range.
- **Memory Limit**: Usually not a deal breaker (unless you are doing something ghastly!).

Example I

Problem: Consider a currency system in which there are notes of seven denominations, namely, Rs. 1, Rs. 2, Rs. 5, Rs. 10, Rs. 50, Rs. 100. If the sum of Rs. N is input, write a program to computer smallest number of notes that will combine to give Rs. N.

Example I

Problem: Consider a currency system in which there are notes of seven denominations, namely, Rs. 1, Rs. 2, Rs. 5, Rs. 10, Rs. 50, Rs. 100. If the sum of Rs. N is input, write a program to computer smallest number of notes that will combine to give Rs. N.
Input: The first line contains an integer T, total number of testcases. Then follow T lines, each line contains an integer N.

Example I

Problem: Consider a currency system in which there are notes of seven denominations, namely, Rs. 1, Rs. 2, Rs. 5, Rs. 10, Rs. 50, Rs. 100. If the sum of Rs. N is input, write a program to computer smallest number of notes that will combine to give Rs. N.
Input: The first line contains an integer T, total number of testcases. Then follow T lines, each line contains an integer N.
Output: Display the smallest number of notes that will combine to give N.

Example I

Problem: Consider a currency system in which there are notes of seven denominations, namely, Rs. 1, Rs. 2, Rs. 5, Rs. 10, Rs. 50, Rs. 100. If the sum of Rs. N is input, write a program to computer smallest number of notes that will combine to give Rs. N.
Input: The first line contains an integer T, total number of testcases. Then follow T lines, each line contains an integer N.
Output: Display the smallest number of notes that will combine to give N.
Constraints: $1 \leq T \leq 1000 \quad 1 \leq N \leq 1000000$

Example II

Example:
Input
3
1200
500
242
Output
12
5
7

Example II

Example:
Input
3
1200
500
242
Output
12
5
7
Let us write a program to solve this!

Practice Sites

Here are a few practice site:

Practice Sites

Here are a few practice site:

- Codeforces

Practice Sites

Here are a few practice site:

- Codeforces
- Codechef

Practice Sites

Here are a few practice site:

- Codeforces
- Codechef
- Topcoder

Practice Sites

Here are a few practice site:

- Codeforces
- Codechef
- Topcoder
- Hacker Rank

Practice Sites

Here are a few practice site:

- Codeforces
- Codechef
- Topcoder
- Hacker Rank
- Hacker Earth

Algorithm and DS

Here are a few learning resources:

Algorithm and DS

Here are a few learning resources:

- Code Monk $\rightarrow$ Link

Algorithm and DS

Here are a few learning resources:

- Code Monk $\rightarrow$ Link
- Top Coder DataScience Tutorial $\rightarrow$ Link

Algorithm and DS

Here are a few learning resources:

- Code Monk $\rightarrow$ Link
- Top Coder DataScience Tutorial $\rightarrow$ Link
- NPTEL Videos *Naveen Garg* (Data structures and Algorithms)

Algorithm and DS

Here are a few learning resources:

- Code Monk $\rightarrow$ Link
- Top Coder DataScience Tutorial $\rightarrow$ Link
- NPTEL Videos *Naveen Garg* (Data structures and Algorithms)
- Contest Tutorials

Language Reference

- Use either C++ or Java . Prefer C++

Language Reference

- Use either C++ or Java . Prefer C++
- C++ Tutorial $\rightarrow$ Link

Language Reference

- Use either C++ or Java . Prefer C++
- C++ Tutorial $\rightarrow$ Link
- STL Tutorial $\rightarrow$ Link

Competitions

- Codeforces Contests $\{Bi\ Weekly\}$

Competitions

- Codeforces Contests {*Bi Weekly*}
- Codechef Long, Lunchtime {*Monthly*}

Competitions

- Codeforces Contests {*Bi Weekly*}
- Codechef Long, Lunchtime {*Monthly*}
- ACM ICPC {*Most Prestigious Programming Contest*}

Competitions

- Codeforces Contests {*Bi Weekly*}
- Codechef Long, Lunchtime {*Monthly*}
- ACM ICPC {*Most Prestigious Programming Contest*}
- Google APAC

Competitions

- Codeforces Contests {*Bi Weekly*}
- Codechef Long, Lunchtime {*Monthly*}
- ACM ICPC {*Most Prestigious Programming Contest*}
- Google APAC
- Facebook Hackercup

Algorithm Class

Basic class of algorithms:{*Increasing order of difficulty*}

Algorithm Class

Basic class of algorithms:{*Increasing order of difficulty*}

- Ad-hoc

Algorithm Class

Basic class of algorithms:{*Increasing order of difficulty*}

- Ad-hoc
- Greedy

Algorithm Class

Basic class of algorithms:{*Increasing order of difficulty*}

- Ad-hoc
- Greedy
- Divide and Conquer

Algorithm Class

Basic class of algorithms:{*Increasing order of difficulty*}

- Ad-hoc
- Greedy
- Divide and Conquer
- Dynamic Programming

Algorithm Class

Basic class of algorithms:{*Increasing order of difficulty*}

- Ad-hoc
- Greedy
- Divide and Conquer
- Dynamic Programming
- Network Flows

Tools Needed

These things are sufficient to practice coding:

Tools Needed

These things are sufficient to practice coding:

- Simple Text Editor. **No IDE !**

Tools Needed

These things are sufficient to practice coding:

- Simple Text Editor. **No IDE !**
- Compiler. GCC is standard

Tools Needed

These things are sufficient to practice coding:

- Simple Text Editor. **No IDE !**
- Compiler. GCC is standard
- Preferably Linux OS (Ubuntu works best)

Tools Needed

These things are sufficient to practice coding:

- Simple Text Editor. **No IDE !**
- Compiler. GCC is standard
- Preferably Linux OS (Ubuntu works best)
- Internet Connection (for problem statements)

Tools Needed

These things are sufficient to practice coding:

- Simple Text Editor. **No IDE !**
- Compiler. GCC is standard
- Preferably Linux OS (Ubuntu works best)
- Internet Connection (for problem statements)
- Online compilers like Ideone , Codechef IDE can also be used. *Beware, they are painfully slow*

Let us look at some basic problems.

Fibonacci Numbers

Write a program to return n^{th} Fibonacci number.
Fibonacci numbers follow the series: $0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$
What is the complexity of your code ?

Fibonacci Numbers

Write a program to return n^{th} Fibonacci number.
Fibonacci numbers follow the series: $0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$
What is the complexity of your code ? Exponential ?

Fibonacci Numbers

Write a program to return n^{th} Fibonacci number.
Fibonacci numbers follow the series: $0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$
What is the complexity of your code ? Exponential ?
If yes, try making it linear

Fibonacci Numbers

Write a program to return n^{th} Fibonacci number.
Fibonacci numbers follow the series: $0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$
What is the complexity of your code ? Exponential ?
If yes, try making it linear
Hint

Fibonacci Numbers

Write a program to return n^{th} Fibonacci number.
Fibonacci numbers follow the series: $0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$
What is the complexity of your code ? Exponential ?
If yes, try making it linear
Hint
Try writing a recurrence relation

Fibonacci Numbers

Write a program to return n^{th} Fibonacci number.
Fibonacci numbers follow the series: $0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$
What is the complexity of your code ? Exponential ?
If yes, try making it linear
Hint
Try writing a recurrence relation
$Fib(0) = 0$

Fibonacci Numbers

Write a program to return n^{th} Fibonacci number.
Fibonacci numbers follow the series: $0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$
What is the complexity of your code ? Exponential ?
If yes, try making it linear
Hint
Try writing a recurrence relation
$Fib(0) = 0$
$Fib(1) = 1$

Fibonacci Numbers

Write a program to return n^{th} Fibonacci number.
Fibonacci numbers follow the series: $0, 1, 1, 2, 3, 5, 8, 13, 21, \ldots$
What is the complexity of your code ? Exponential ?
If yes, try making it linear

Hint

Try writing a recurrence relation

$Fib(0) = 0$
$Fib(1) = 1$
$Fib(n) = Fib(n-1) + Fib(n-2) \quad , n \geq 2$

Exponentiation

Write a program to calculate a^b given a and b as inputs. Both are natural numbers and the result is expressible in *int* range.

Exponentiation

Write a program to calculate a^b given a and b as inputs. Both are natural numbers and the result is expressible in *int* range.
Did you calculate it as: $\underbrace{a * a * \ldots * a}_{b\text{-times}}$

Exponentiation

Write a program to calculate a^b given a and b as inputs. Both are natural numbers and the result is expressible in *int* range.
Did you calculate it as: $\underbrace{a * a * \ldots * a}_{b\text{-times}}$

Can you do better?

Exponentiation

Write a program to calculate a^b given a and b as inputs. Both are natural numbers and the result is expressible in *int* range.
Did you calculate it as: $\underbrace{a * a * \ldots * a}_{b\text{-times}}$

Can you do better?
Of Course!

Exponentiation

Write a program to calculate a^b given a and b as inputs. Both are natural numbers and the result is expressible in *int* range.
Did you calculate it as: $\underbrace{a * a * \ldots * a}_{b\text{-times}}$

Can you do better?
Of Course!
Use Fast Exponentiation!

Interval Selection

This is a harder problem than the above ones.

Interval Selection

This is a harder problem than the above ones.
You are given n activities with their start and finish times. Select the maximum number of activities that can be performed by a single person, assuming that a person can only work on a single activity at a time.

Interval Selection

This is a harder problem than the above ones.
You are given n activities with their start and finish times. Select the maximum number of activities that can be performed by a single person, assuming that a person can only work on a single activity at a time.
Try doing it on your own. If nothing comes, then Google!

Important Notes

Here are some general closing notes:

Important Notes

Here are some general closing notes:

- Practice Regularly.

Important Notes

Here are some general closing notes:

- Practice Regularly.
- Make a group of enthusiastic people to participate in contests.

Important Notes

Here are some general closing notes:

- Practice Regularly.
- Make a group of enthusiastic people to participate in contests.
- Practice Regularly.

Thank You