ICPC Sessions
OR
How to Solve Problems

Sebastian Claici
sebastianclaici@gmail.com

November 29, 2012

What is the ACM ICPC?

What is the ACM ICPC?

- The ACM International Collegiate Programming Contest!

What is the ACM ICPC?

- The ACM International Collegiate Programming Contest!
- Team based (teams of 3)

What is the ACM ICPC?

- The ACM International Collegiate Programming Contest!
- Team based (teams of 3)
- The most prestigious global programming competition (since 1977)!

This is how it goes...

This is how it goes...

- Regionals and Finals

This is how it goes...

- Regionals and Finals
- We are part of the Northwestern European region

This is how it goes...

- Regionals and Finals
- We are part of the Northwestern European region
- Top 3 teams will qualify for the Finals in St. Petersburg

This is how it goes...

- Regionals and Finals
- We are part of the Northwestern European region
- Top 3 teams will qualify for the Finals in St. Petersburg
- This means we will make Germany, Belgium, the Netherlands and others cry!

What is this about?

What is this about?

- Algorithms; more than you did in previous years (if any)!

What is this about?

- Algorithms; more than you did in previous years (if any)!
- Actually using algorithms to solve problems!

What is this about?

- Algorithms; more than you did in previous years (if any)!
- Actually using algorithms to solve problems!
- Beating Cambridge (and everyone else)!

Why study this?

Why study this?

Google™

Why study this?

Google™

facebook

Why study this?

Google™

facebook

Microsoft

List of almost everything

List of almost everything

- Ad-hoc

List of almost everything

- Ad-hoc
- Common data structures (stacks, queues, maps, etc.)

List of almost everything

- Ad-hoc
- Common data structures (stacks, queues, maps, etc.)
- Problem solving paradigms

List of almost everything

- Ad-hoc
- Common data structures (stacks, queues, maps, etc.)
- Problem solving paradigms
    - Complete search

List of almost everything

- Ad-hoc
- Common data structures (stacks, queues, maps, etc.)
- Problem solving paradigms
    - Complete search
    - Divide and conquer

List of almost everything

- Ad-hoc
- Common data structures (stacks, queues, maps, etc.)
- Problem solving paradigms
    - Complete search
    - Divide and conquer
    - Greedy

List of almost everything

- Ad-hoc
- Common data structures (stacks, queues, maps, etc.)
- Problem solving paradigms
    - Complete search
    - Divide and conquer
    - Greedy
    - Dynamic programming

List of almost everything

- Ad-hoc
- Common data structures (stacks, queues, maps, etc.)
- Problem solving paradigms
  - Complete search
  - Divide and conquer
  - Greedy
  - Dynamic programming
- Graphs

List of almost everything

- Ad-hoc
- Common data structures (stacks, queues, maps, etc.)
- Problem solving paradigms
    - Complete search
    - Divide and conquer
    - Greedy
    - Dynamic programming
- Graphs
- More graphs

List of almost everything

- Ad-hoc
- Common data structures (stacks, queues, maps, etc.)
- Problem solving paradigms
  - Complete search
  - Divide and conquer
  - Greedy
  - Dynamic programming
- Graphs
- More graphs
- Even more graphs

List of almost everything

- Ad-hoc
- Common data structures (stacks, queues, maps, etc.)
- Problem solving paradigms
    - Complete search
    - Divide and conquer
    - Greedy
    - Dynamic programming
- Graphs
- More graphs
- Even more graphs
- Maths

List of almost everything

- Ad-hoc
- Common data structures (stacks, queues, maps, etc.)
- Problem solving paradigms
    - Complete search
    - Divide and conquer
    - Greedy
    - Dynamic programming
- Graphs
- More graphs
- Even more graphs
- Maths
- Computational geometry

List of almost everything

- Ad-hoc
- Common data structures (stacks, queues, maps, etc.)
- Problem solving paradigms
  - Complete search
  - Divide and conquer
  - Greedy
  - Dynamic programming
- Graphs
- More graphs
- Even more graphs
- Maths
- Computational geometry
- String processing

How to Prepare

How to Prepare

- Beginners:

How to Prepare

- Beginners:
  - USACO, UVa, Project Euler (last one not recommended)

How to Prepare

- Beginners:
  - USACO, UVa, Project Euler (last one not recommended)
- Intermediate:

How to Prepare

- Beginners:
  - USACO, UVa, Project Euler (last one not recommended)
- Intermediate:
  - `http://poj.org`, `http://acm.tju.ed.cn`, `http://acm.sgu.ru`, `http://acm.timus.ru`

How to Prepare

- Beginners:
  - USACO, UVa, Project Euler (last one not recommended)
- Intermediate:
  - `http://poj.org`, `http://acm.tju.ed.cn`, `http://acm.sgu.ru`, `http://acm.timus.ru`
- Everyone:

How to Prepare

- Beginners:
  - USACO, UVa, Project Euler (last one not recommended)
- Intermediate:
  - `http://poj.org`, `http://acm.tju.ed.cn`, `http://acm.sgu.ru`, `http://acm.timus.ru`
- Everyone:
  - Topcoder and Codeforces - make accounts on both

How to Prepare

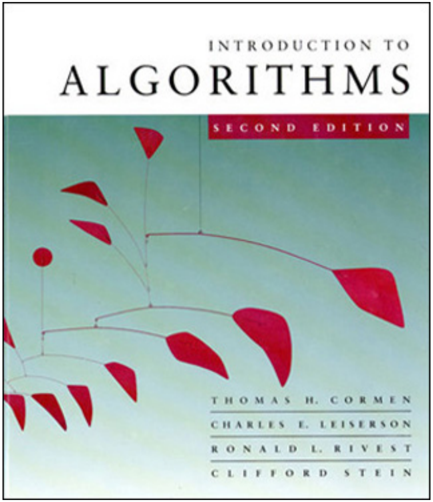
- Beginners:
  - USACO, UVa, Project Euler (last one not recommended)
- Intermediate:
  - http://poj.org, http://acm.tju.ed.cn, http://acm.sgu.ru, http://acm.timus.ru
- Everyone:
  - Topcoder and Codeforces - make accounts on both
- Romanians:

How to Prepare

- Beginners:
    - USACO, UVa, Project Euler (last one not recommended)
- Intermediate:
    - `http://poj.org`, `http://acm.tju.ed.cn`, `http://acm.sgu.ru`, `http://acm.timus.ru`
- Everyone:
    - Topcoder and Codeforces - make accounts on both
- Romanians:
    - Infoarena

Recommended Book(s)

- **Introduction to Algorithms**
  - Thomas Cormen, Charles Leiserson, Ronald Rivest, Clifford Stein
- **Algorithms in C/C++/Java/**
  - Robert Sedgewick


INTRODUCTION TO
ALGORITHMS
SECOND EDITION
THOMAS H. CORMEN
CHARLES E. LEISERSON
RONALD L. RIVEST
CLIFFORD STEIN

Challenge

Challenge

90% of programmers, given 2 hours, the high-level language of their choice (including pseudocode), and a description of binary search could not implement it correctly.

Challenge

90% of programmers, given 2 hours, the high-level language of their choice (including pseudocode), and a description of binary search could not implement it correctly.

Can you?

Binary Search

```c
int binary_search(int *array, int n, int x)
{
    int lo = 0, hi = n - 1;
    while (lo < hi) {
        int mid = lo + (hi - lo) / 2;
        if (array[mid] < x)
            lo = mid + 1;
        else hi = mid;
    }

    if (lo == hi && array[lo] == x)
        return lo;
    return -1;
}
```

Ad-hoc Problems

Characteristics

Characteristics

- Most of them are very easy; some of them are very very hard

Characteristics

- Most of them are very easy; some of them are very very hard
- Don't require any special knowledge of algorithms

Characteristics

- Most of them are very easy; some of them are very very hard
- Don't require any special knowledge of algorithms
- There is always at least one in competitions

Strategies to solve

Strategies to solve

- Most of these problems are straightforward.

Strategies to solve

- Most of these problems are straightforward.
- However, some ad-hoc problems require careful reading.

Strategies to solve

- Most of these problems are straightforward.
- However, some ad-hoc problems require careful reading.
- Carefully sequencing the instructions given in the problem is usually enough to solve them.

Strategies to solve

- Most of these problems are straightforward.
- However, some ad-hoc problems require careful reading.
- Carefully sequencing the instructions given in the problem is usually enough to solve them.
- Some require reasonable optimisations, and some degree of analysis to prune unnecessary steps.

Strategies to solve

- Most of these problems are straightforward.
- However, some ad-hoc problems require careful reading.
- Carefully sequencing the instructions given in the problem is usually enough to solve them.
- Some require reasonable optimisations, and some degree of analysis to prune unnecessary steps.
- If it's not obvious, then there's only one piece of advice I can give you:

Strategies to solve

- Most of these problems are straightforward.
- However, some ad-hoc problems require careful reading.
- Carefully sequencing the instructions given in the problem is usually enough to solve them.
- Some require reasonable optimisations, and some degree of analysis to prune unnecessary steps.
- If it's not obvious, then there's only one piece of advice I can give you:

Don't Panic!

Practice, practice, practice...

Practice, practice, practice...

- To get really good at this, you need practice. **A lot** of practice.

Practice, practice, practice...

- To get really good at this, you need practice. **A lot** of practice.
- So I've decided that every week I'll give you a set of **a lot** of problems so that you can practice.

Practice, practice, practice...

- To get really good at this, you need practice. **A lot** of practice.
- So I've decided that every week I'll give you a set of **a lot** of problems so that you can practice.
- Will usually be more than you can solve in a week, although I'd love to be proven wrong.

Practice, practice, practice...

- To get really good at this, you need practice. **A lot** of practice.
- So I've decided that every week I'll give you a set of **a lot** of problems so that you can practice.
- Will usually be more than you can solve in a week, although I'd love to be proven wrong.
- Solutions should be available one week after the problems were set.

Practice, practice, practice...

- To get really good at this, you need practice. **A lot** of practice.
- So I've decided that every week I'll give you a set of **a lot** of problems so that you can practice.
- Will usually be more than you can solve in a week, although I'd love to be proven wrong.
- Solutions should be available one week after the problems were set.

For this week, try your hand at these questions (all ad-hoc):
http://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=121

More Info

More Info

- You can email me about anything related to this. I'll usually respond within a day or two.

More Info

- You can email me about anything related to this. I'll usually respond within a day or two.
- All the slides, and everything we're going to do in these sessions, as well as solutions to weekly problems are on github:

  github.com/sebastian-claici/acm_sessions.git

More Info

- You can email me about anything related to this. I'll usually respond within a day or two.
- All the slides, and everything we're going to do in these sessions, as well as solutions to weekly problems are on github:

  github.com/sebastian-claici/acm_sessions.git

Thank You