Testing
2013 IOI Camp 1

Robert Spencer

December 11, 2013

Testing is one of the most important and most overlooked aspects of olympiads.

Testing is one of the most important and most overlooked aspects of olympiads.
More important in training (detailed feedback).

Introduction

Testing is one of the most important and most overlooked aspects
of olympiads.
More important in training (detailed feedback).
Also useful for finding and fixing bugs.

Testing is one of the most important and most overlooked aspects of olympiads.

More important in training (detailed feedback).

Also useful for finding and fixing bugs.

Test by giving your program multiple inputs before submitting (if time is no object).

Basics

Test by giving your program multiple inputs before submitting (if time is no object).
Here multiple test cases in files are very useful.

Test by giving your program multiple inputs before submitting (if time is no object).
Here multiple test cases in files are very useful.

```
program < testcase
```

Test by giving your program multiple inputs before submitting (if time is no object).

Here multiple test cases in files are very useful.

```
program < testcase
```

Have a slow brute force that you can compare answers against.

Test by giving your program multiple inputs before submitting (if time is no object).
Here multiple test cases in files are very useful.

```
program < testcase
```

Have a slow brute force that you can compare answers against.
Use printing (even if you are not debugging) and asserts.

Write sample input makers. Here scripting languages are useful.

Makers

Write sample input makers. Here scripting languages are useful.

```
import random
print 1000
for i in range(1000):
  print random.randint(1,1000000),
```

Write sample input makers. Here scripting languages are useful.

```
import random
print 1000
for i in range(1000):
  print random.randint(1,1000000),
```

And then use files:

```
$ python maker.py > input
$ program < input > output
($ bruteforce < input > bruteoutput
 $ diff output bruteoutput)
```

Corner Cases

Don't forget corner cases:

- $N = 0, 1$
- $M = N$
- All values the same
- All values different
- Values not consecutive (!)

Corner Cases

Don't forget corner cases:

- $N = 0, 1$
- $M = N$
- All values the same
- All values different
- Values not consecutive (!)

"There are two things I hate in life:

Corner Cases

Don't forget corner cases:

- $N = 0, 1$
- $M = N$
- All values the same
- All values different
- Values not consecutive (!)

"There are two things I hate in life:off by one errors"

Try to use the full range of the constraints.

Try to use the full range of the constraints.
If $1 \leq N \leq 10^6$, make a case where $N = 10^6$.

Use the server given testing facilties.

Use the server given testing facilties.
The server may behave slightly differently (memory bugs, time).

Use the server given testing facilties.
The server may behave slightly differently (memory bugs, time).
At IOI, you get a lot of data back from the server.

The Common Pitfalls

Most people test far to little. In general, unless you know that your solution is correct and cannot at all break it, you should not be satisfied that you have tested enough.