Project 4
Search Based Testing

Bit Representation

- Java uses two's complement.
  - 1 ⇒ 00000000000000000000000000000001
  - -1 ⇒ 11111111111111111111111111111111

Flipping Bits

- Flipping most significant bit (sign bit).
- 11111111111111111111111111111111
- 01111111111111111111111111111111
- -1 becomes Integer.MAX_VALUE.

Flipping Bits

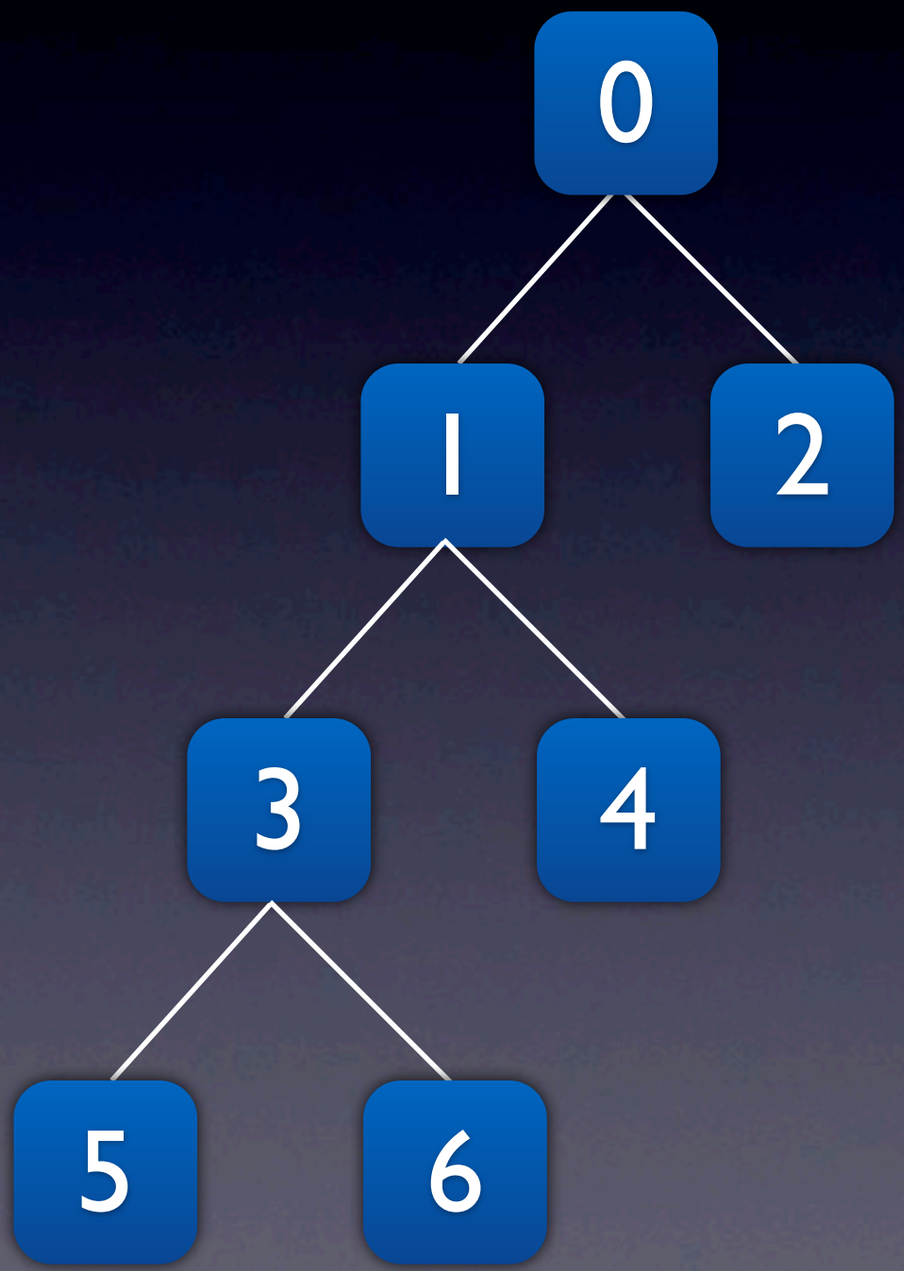
- Might cause trouble when a value >0 is needed.
- Algorithm approaches from negative side.
- Every positive value is disregarded because of its fitness.
- Depends on parameters.
- Possible solution: Multiply with -1 instead of flipping most significant bit.

Dependencies

```
m l (int a,int b)
if(a > 10){
   if(b > 20){
      if(a + b > 50){
      }
   }
}
```

Expression	TRUE	FALSE
if(a > 10)	1	2
if(b > 20)	3	4
if(a + b > 50)	5	6

Dependencies



Expression	TRUE	FALSE
if(a > 10)	1	2
if(b > 20)	3	4
if(a + b > 50)	5	6

Secret Tests

```
boolean a=true;
boolean b=true;
if (a && b) {
...
```

- Not covered by public tests
- Boolean variables
- Methods that return booleans
- Boolean literals

Boolean Expressions

Table 1: Distance metrics

Construct	Metric
$a = b$	$a - b = 0\ ?\ 0 : abs(a - b) + k$
$a \neq b$	$a - b \neq 0\ ?\ 0 : k$
$a < b$	$a - b < 0\ ?\ 0 : (a - b) + k$
$a \leq b$	$a - b \leq 0\ ?\ 0 : (a - b) + k$
$a > b$	$b - a < 0\ ?\ 0 : (b - a) + k$
$a \geq b$	$b - a \leq 0\ ?\ 0 : (b - a) + k$
boolean	$true?\ 0 : k$
a && b	$distance(a) + distance(b)$
$a \mid\mid b$	$min(distance(a), distance(b))$
$!a$	Move inward and propagate, e.g $!(a > b)$ becomes $a \leq b$ and $!(a \&\& b)$ becomes $!a \mid\mid !b$.

distance for a && b

distance = ((a) ? 1 : 0) + ((b) ? 1 : 0)

Secret Tests

```
boolean a=true;
boolean b=true;
boolean c=i > 6;
if (a && b && c) {
...
```

- Extended Operants.

Pitfalls

1 + 2

vs.

1 + 2 + 3

Pitfalls

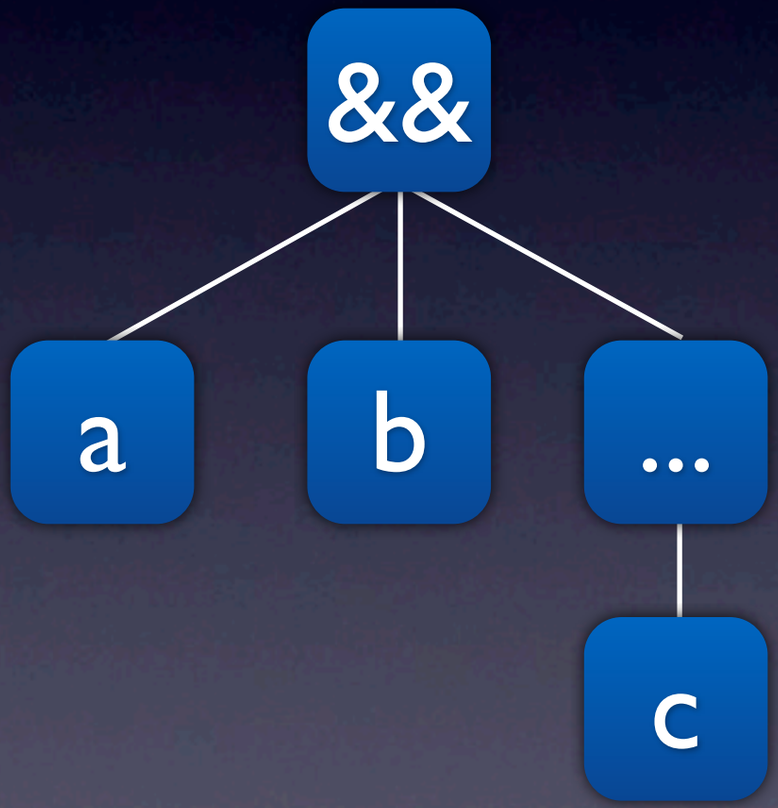
a && b

vs.

a && b && c

Pitfalls

AST



a && b

a && b && c

Pitfalls

a && b

vs.

a && b && c

- AstParser handles these expressions differently.
- See node.extendedOperands()