

CS367 Announcements

Wednesday, June 26, 2013

- H2 released, due Monday July 1st, 6pm
- P1 due in a week
- H1 grading
- no class July 4th

Last Time

- Complexity

Today

- Complexity (cont.)

Complexity Review

- **problem size**
- **count**
- As **problem size** grows, how does **count** scale? (usually in worst case)
- Use: comparison of algorithms
- **Big-O**: least upper bound
- **order** classes we're interested in
- **dominating term**

Complexity in Java Code

Basic operations

Sequence of statements

```
statement1;  
statement2;  
...  
statementk;
```

If-then-else

```
if (cond) {  
    sequence1 of statements  
} else {  
    sequence2 of statements  
}
```

Complexity in Java Code

Loops

```
for (i = 0; i < N; i++) {  
    sequence of statements  
}
```

Nested Loops

```
for (i = 0; i < N; i++) {  
    for (j = 0; j < M; j++) {  
        sequence of statements  
    }  
}
```

Method calls

```
for (i = 0; i < N; i++) {  
    f(i);  
}
```

```
for (i = 0; i < N; i++) {  
    g(N);  
}
```

```
for (i = 0; i < N; i++) {  
    g(i);  
}
```

Returning N papers to N students

problem size (N) =

count number of =

What is the complexity of each algorithm below?

Algorithm 1: call out each name, have student come forward & pick up

best-case:

worst-case:

Algorithm 2: hand pile to first student, student searches through papers & takes hers/his, pass pile to next student

best-case:

worst-case:

Algorithm 3: sort the papers alphabetically, hand pile to first student, they do binary search & pass to next student

best-case:

worst-case: