# CS350: Data Structures Class Notes

*Class 9: Friday, September 19, 2008*

**Complexity**

This class devoted to complexity. We looked at a table showing how expressions grow:

```
f[x_] := 2 x^2 + 10 x + 100
g[x_] := 2 x^2
Table[{10^i, f[10^i], g[10^i]}, {i,  0, 10}] // MatrixForm
```

| | | |
|---|---|---|
| 1 | 112 | 2 |
| 10 | 400 | 200 |
| 100 | 21100 | 20000 |
| 1000 | 2010100 | 2000000 |
| 10000 | 200100100 | 200000000 |
| 100000 | 20001000100 | 20000000000 |
| 1000000 | 2000010000100 | 2000000000000 |
| 10000000 | 200000100000100 | 200000000000000 |
| 100000000 | 20000001000000100 | 20000000000000000 |
| 1000000000 | 2000000010000000100 | 2000000000000000000 |
| 10000000000 | 200000000100000000100 | 200000000000000000000 |

```
g[x_] := 2 x^2
h[x_] := x^2
Table[{10^i, g[10^i], h[10^i]}, {i,  0, 10}] // MatrixForm
```

| | | |
|---|---|---|
| 1 | 2 | 1 |
| 10 | 200 | 100 |
| 100 | 20000 | 10000 |
| 1000 | 2000000 | 1000000 |
| 10000 | 200000000 | 100000000 |
| 100000 | 20000000000 | 10000000000 |
| 1000000 | 2000000000000 | 1000000000000 |
| 10000000 | 200000000000000 | 100000000000000 |
| 100000000 | 20000000000000000 | 10000000000000000 |
| 1000000000 | 2000000000000000000 | 1000000000000000000 |
| 10000000000 | 200000000000000000000 | 100000000000000000000 |

It should be clear that the power term dominates. (These tables were generated with *Mathematica*® with output reformatted slightly.)

The next example show that, eventually, the coefficient doesn't matter as n goes from 10 to 11 and 100 to 101.

```
{2*10.0^20, 11.0^20, 2*100.0^75, 101.0^75} →
   {2.*10^20, 6.7275*10^20, 2.*10^150, 2.10913*10^150}
```

We looked at the definitions for and distinctions among $O(n)$ (upper bound), $\Omega(n)$ (lower bound) and $\Theta(n)$ tight bound. We saw that for a polynomial, they are the same.

Growth hierarchy: $1, \log \log n, \log, n, n \log n, n^2, n^3, 2^n, 10^n, n!, n^n$

**No homework for Monday other than reading.**