CSE 421
Algorithms

Richard Anderson

Lecture 12

Recurrences

Announcements

- Wednesday class will meet in CSE 305.

Divide and Conquer

```
Array Mergesort(Array a){
        n = a.Length;
        if (n <= 1)
                return a;
        b = Mergesort(a[0..n/2]);
        c = Mergesort(a[n/2+1 .. n-1]);
        return Merge(b, c);
```

Algorithm Analysis

- Cost of Merge
- Cost of Mergesort

$T(n) = 2T(n/2) + cn; T(1) = c;$

Recurrence Analysis

- Solution methods
  - – Unrolling recurrence
  - – Guess and verify
  - – Plugging in to a “Master Theorem”

A better mergesort (?)

- Divide into 3 subarrays and recursively sort
- Apply 3-way merge

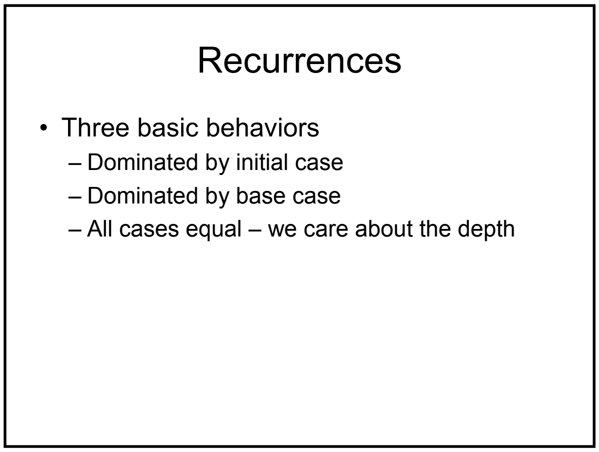
$T(n) = aT(n/b) + f(n)$

$T(n) = T(n/2) + cn$

$T(n) = 4T(n/2) + cn$

$T(n) = 2T(n/2) + n^2$

$T(n) = 2T(n/2) + n^{1/2}$



Recurrences

- Three basic behaviors
  - Dominated by initial case
  - Dominated by base case
  - All cases equal – we care about the depth