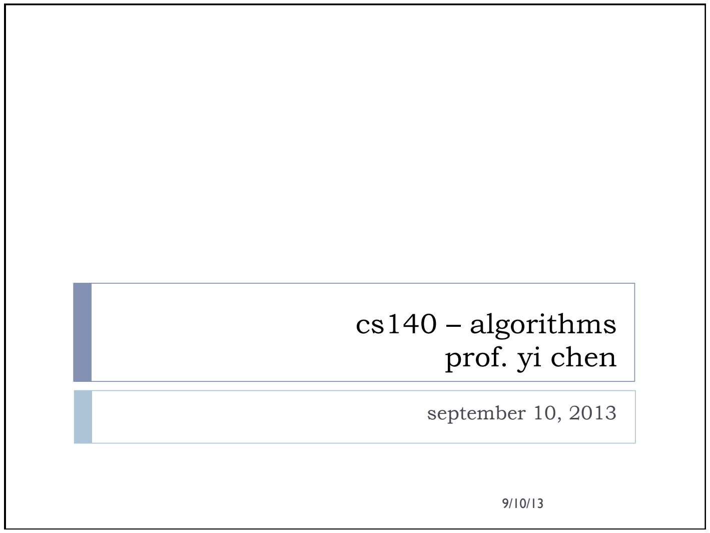
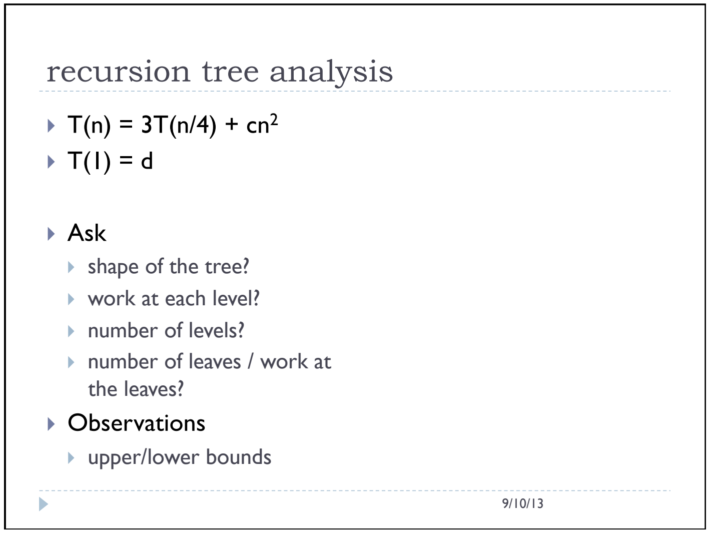


cs140 – algorithms
prof. yi chen

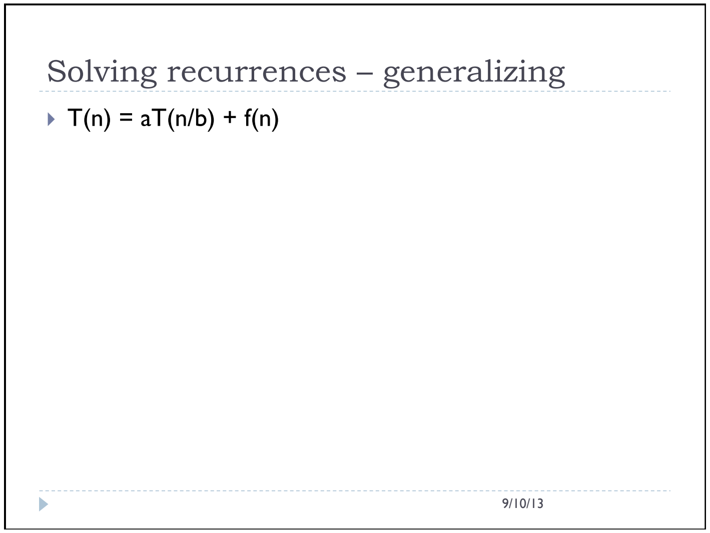
september 10, 2013



recursion tree analysis

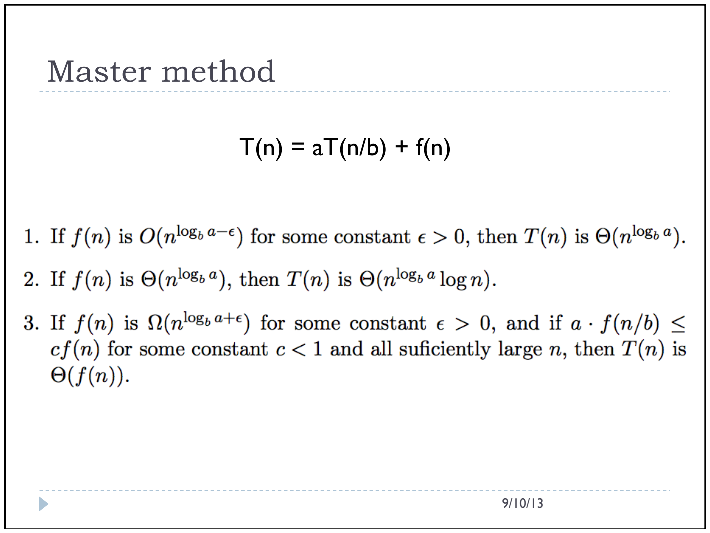
- T(n) = 3T(n/4) + cn^2
- T(1) = d

- Ask
  - shape of the tree?
  - work at each level?
  - number of levels?
  - number of leaves / work at the leaves?
- Observations
  - upper/lower bounds



Solving recurrences – generalizing

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



Master method

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

1. If $f(n)$ is $O(n^{\log_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n)$ is $\Theta(n^{\log_b a})$.
2. If $f(n)$ is $\Theta(n^{\log_b a})$, then $T(n)$ is $\Theta(n^{\log_b a} \log n)$.
3. If $f(n)$ is $\Omega(n^{\log_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $a \cdot f(n/b) \leq cf(n)$ for some constant $c < 1$ and all suficiently large n, then $T(n)$ is $\Theta(f(n))$.

Quicksort - divide-and-conquer

- divide the problem into a number of subproblems.
- conquer by solving the subproblems (do this recursively)
- combine the solutions to the subproblems to give a solution to the overall problem.

Quicksort

```
quicksort(A,p,r)
    if p<r
        q = partition(A,p,r)
        quicksort(A,p,q-1)
        quicksort(A,q+1,r)
    endif
```

```
partition(A,p,r)
    pivot = A[r]
    i = p-1
    for j=p:r
        if A[j] <= pivot
            i=i+1
            swap A[i] and A[j]
        endif
    endfor
    swap A[r] and A[i+1]
    return i+1
```

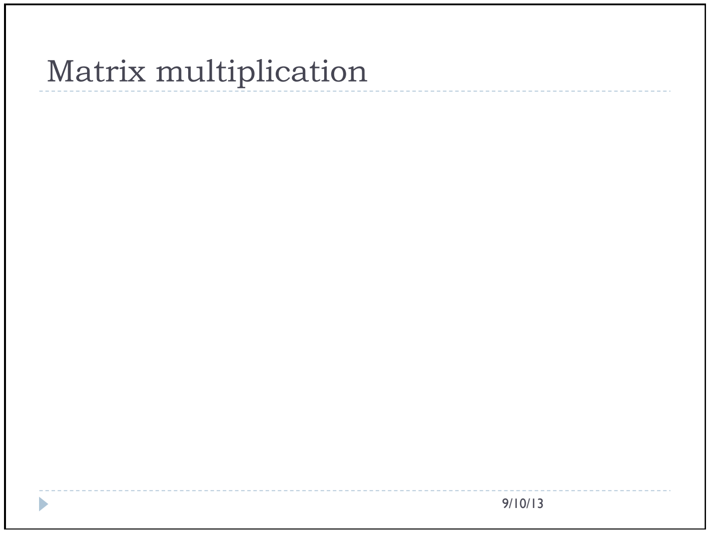
- correctness?
- running time?

Quicksort – unbalanced partition?

- Always splits 4 to 1?

Bounding algorithms vs. problems

- algorithms
  - best case, average case, worst case
  - for each: upper bound (O), lower bound (Ω), tight bound (Θ)
- problems
  - best possible algorithm?



Matrix multiplication

9/10/13