① Syllabus changes and review

② Loop invariants for iterative algorithms → Initialization
→ Maintenance
→ Termination

③ Recursive insertion sort

```
Rec-Ins-Sort (A, 1, n)
    If (n=1)
        return
    else
        Rec-Ins-Sort (A, 1, n-1)
        t = A[n]
        r = A[n-1];
        while (t < A[r] and r > 0)
            A[r+1] = A[r]
            r = r-1
        endwhile
        A[r+1] = t
```

$T(1) = 0$

$T(n) = T(n-1) + n$

```
Rec-Ins (A, 1, k, key)
    If A[k] <= key
        A[k+1] = key
    else
        A[k+1] = A[k]
        Rec-Ins (A, 1, k-1, key)
```

If (k = 1)

① Ajwan, ② Grajen, ③

Asymptotics

- Osc slides
- Monotonicity ; strict monotonicity
- Floors ceilings
- exponent laws

$$n^b = o(a^n)$$

- Logs $\lg n$, $\ln n$, $\lg^k n$, $\lg\lg n$

$$a = b^{\log_b a} \quad \log ab, \quad \log a/b, \quad \log_b a^n$$

$$\log^b n = o(n^a)$$

- Function iteration

$$f^{(i)}(n) = \begin{cases} n, & \text{if } i = 0 \\ f(f^{(i-1)}(n)) & i > 0 \end{cases}$$

e.g. $f(n) = 2n \rightarrow f^{(i)}(n) = 2^i n$

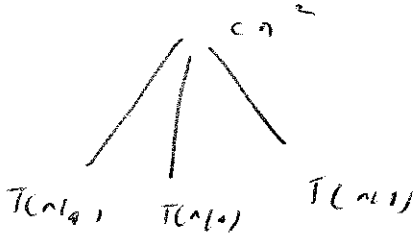
$$\lg^* n = \min\{i \geq 0 : \lg^{(i)} n \leq 1\}$$

$$\lg^*(2^{65536}) = 5$$

Recursion tree method

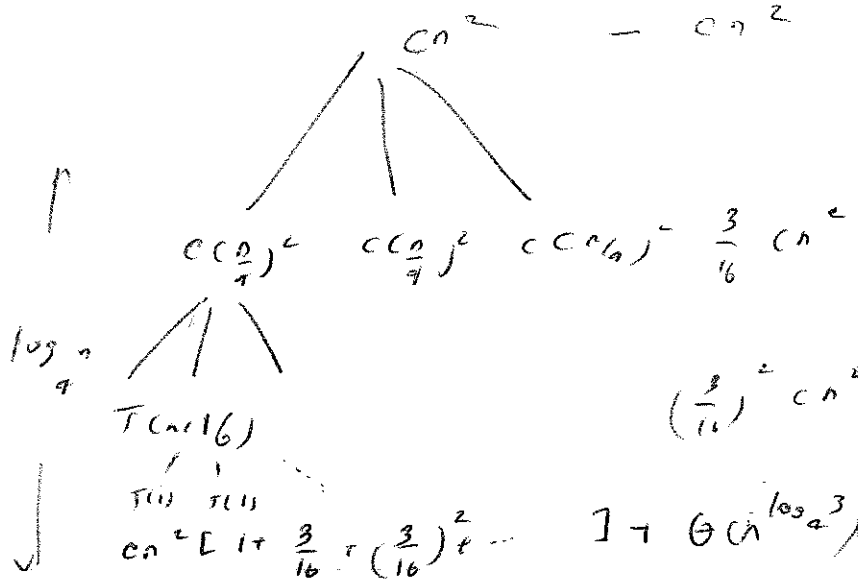
$T(1) = 1$

$T(n) = 3T(n/4) + cn^2$



$T(1) = 0$

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



$cn^2 [1 + \frac{3}{16} + (\frac{3}{16})^2 + \dots] + \Theta(n^{\log_4 3})$

$= O(n^2)$

Each level has 3 more than above.

$3^{\log_4 n} = n^{\log_4 3}$

The master method

→ Does not cover all cases