Midterm Review

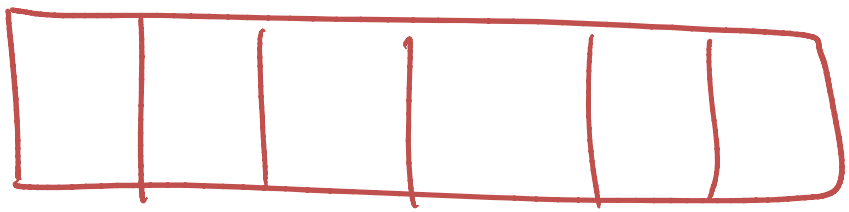
Topics

- Arrays – sorted, unsorted
- LL's – sorted, unsorted, singly, doubly, multi
- Algorithm complexity
- Stacks and queues and applications
- Recursion
- Binary Search Trees

Exam Format

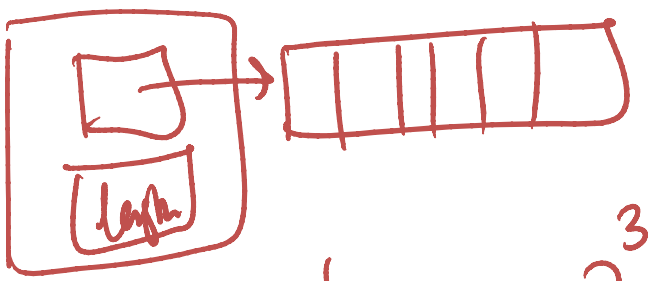
- 80 minutes
- Written , in class exam
- One page of cheat sheet allowed
- No code examples
- Types of questions
  - Short answers
  - Write/trace methods
  - Recursion on java with helper methods
  - Algorithm analysis

Arrays



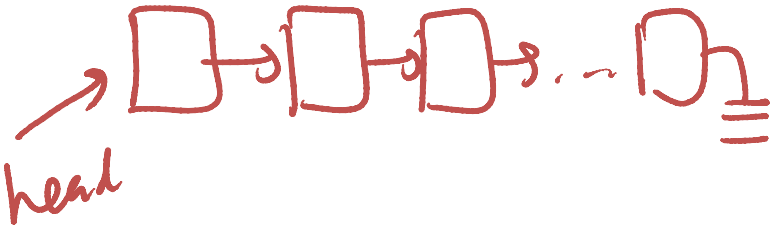
Operations	big O
insert	
delete	
Search	
Sort	
Set an element	
Reverse	
Resize	

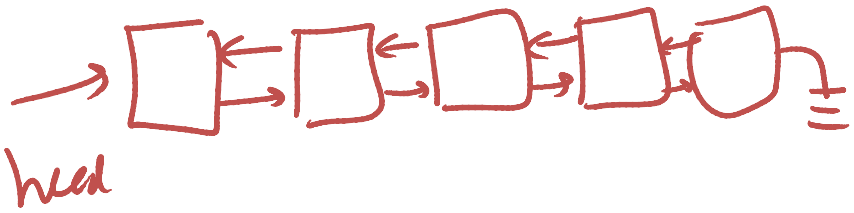
Contiguous
Random access
fixed size
homogeneous
primitive types or Objects



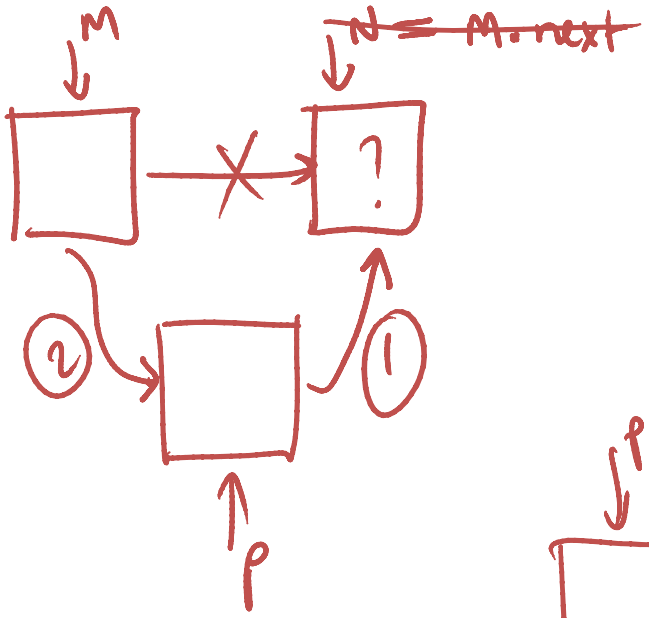
largest signed $= 2^{31} - 1$

LL's



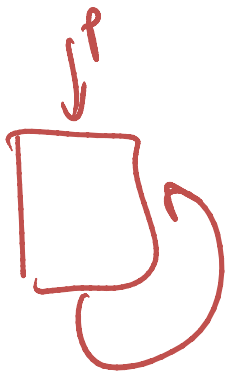


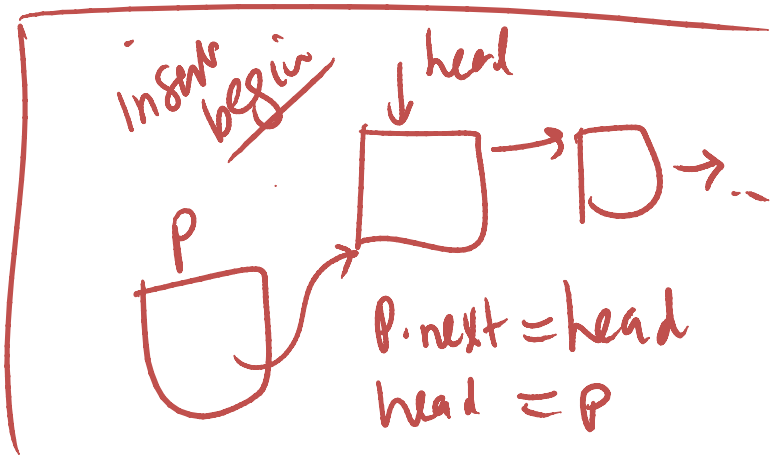
insert

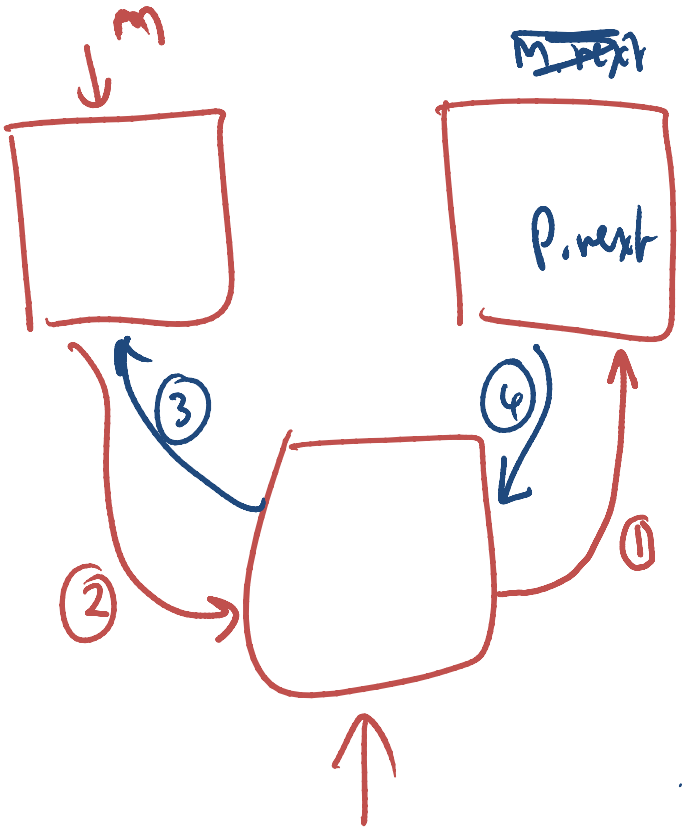


2. M.next = P

1. P.next = M.next







① : P.next = M.next

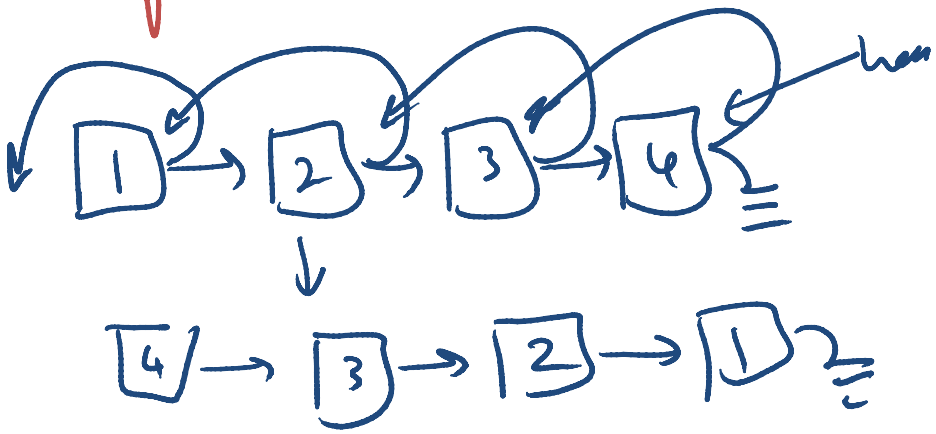
② : M.next = P

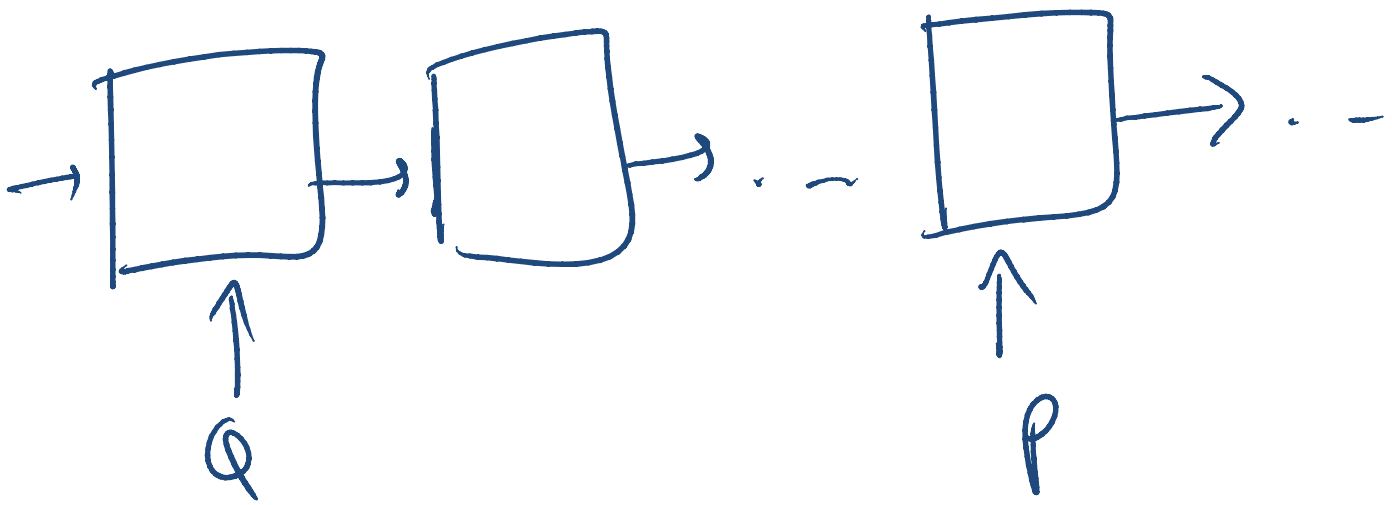
③ : P.prev = M

④ : (P.next).prev = P

Operations

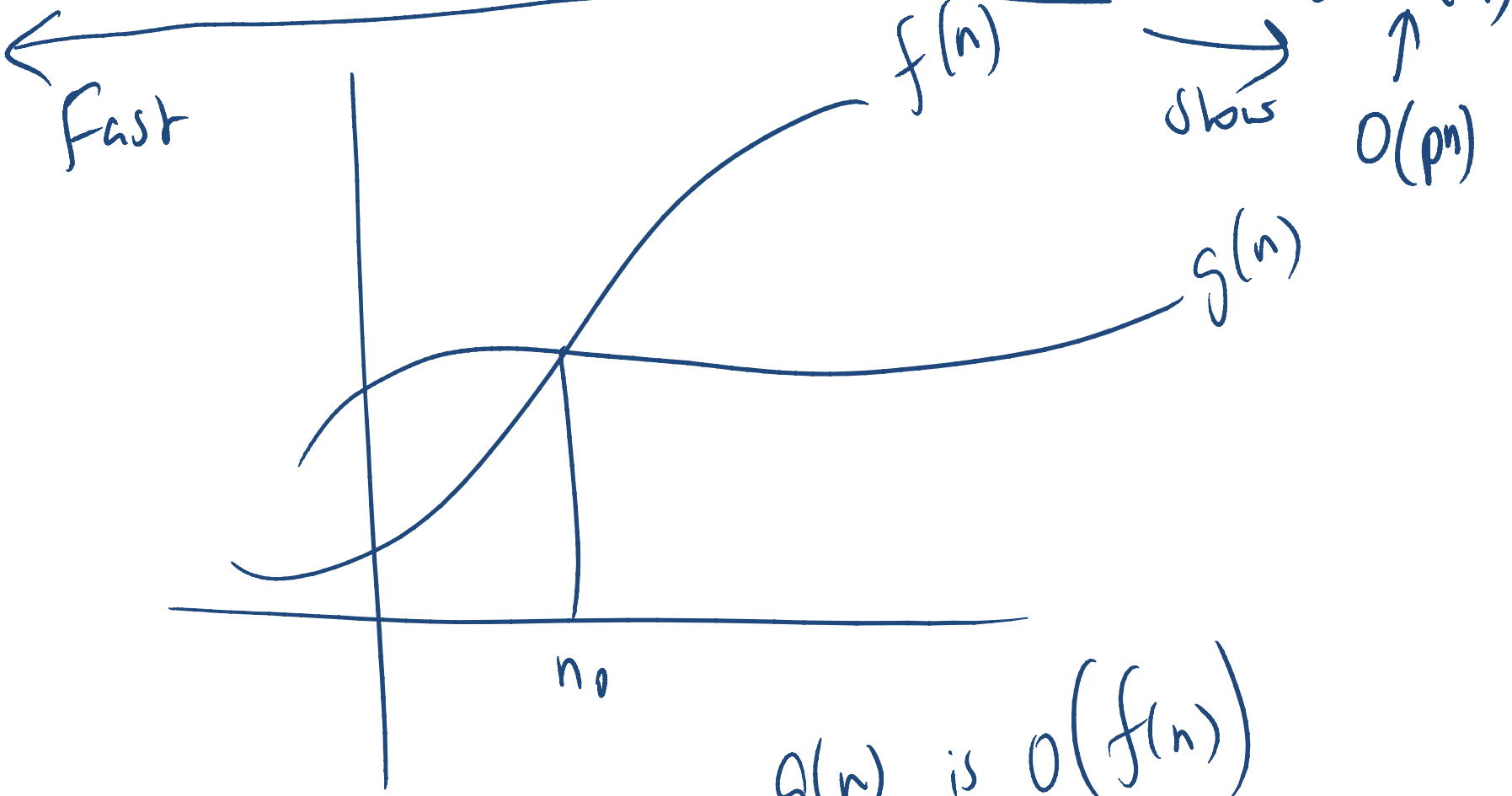
1. Insert Front
2. Insert
3. delete Front
4. delete
5. Find
6. Reverse →





swap(p, q) → ??

$O(1) < O(\log n) < O(n) < O(n \log n) < O(n^2) < O(n^p) < O(n!)$



$g(n)$ is $O(f(n))$

if $g(n) \leq f(n)$

for all $n \geq n_0$

i++=2

```
for (i=1; i<n; i++)
    for (j=i; j<n; j++)
```

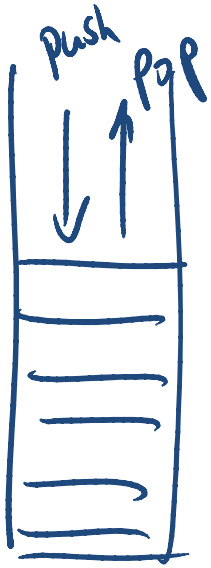
$O(n^2)$

Operation

```
for (j=1; j<n; j*=2)
```

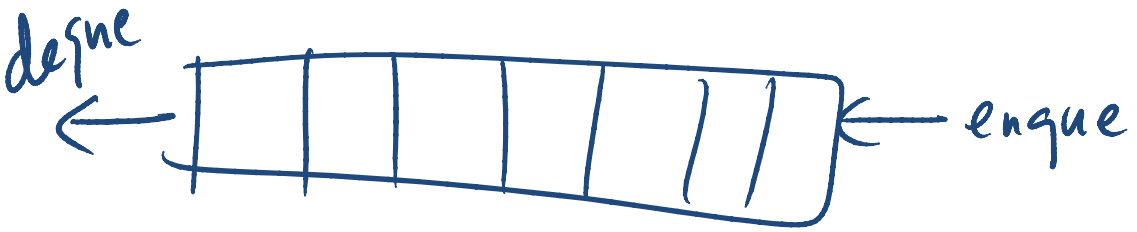
$\longrightarrow O(\log n)$

Stacks | Queues



→ maze traversal (back tracking algorithm)

postfix eval



Recursion

$f(n) = n + f(n-1)$ $\quad f(0) = 0$

$f(n) = n \cdot f(n-1)$ $\quad f(0) = 1$

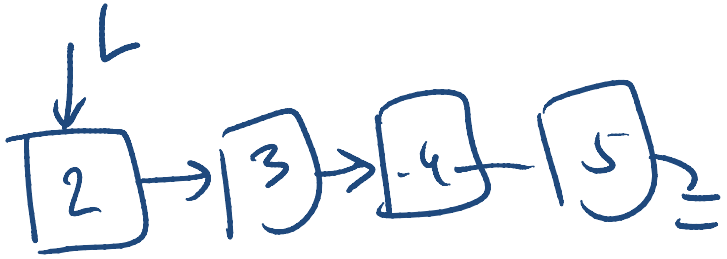
$f(n) = 2 \cdot f(n-1)$ $\quad f(0) = 1$

$f(n) = f(n-1) + f(n-2)$ $\quad f(0) = f(1) = 1$

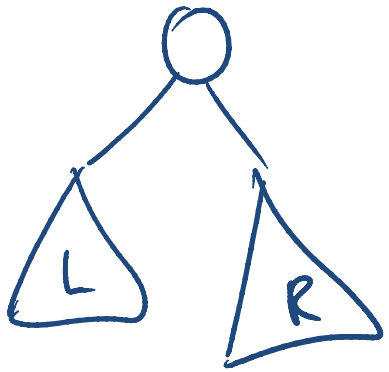
$n \geq 2$

Recursive Method

```
void foo(List L){
    if (L != null){
        System.out.println(L.);
        foo(L.next)
    }
}
```



BST

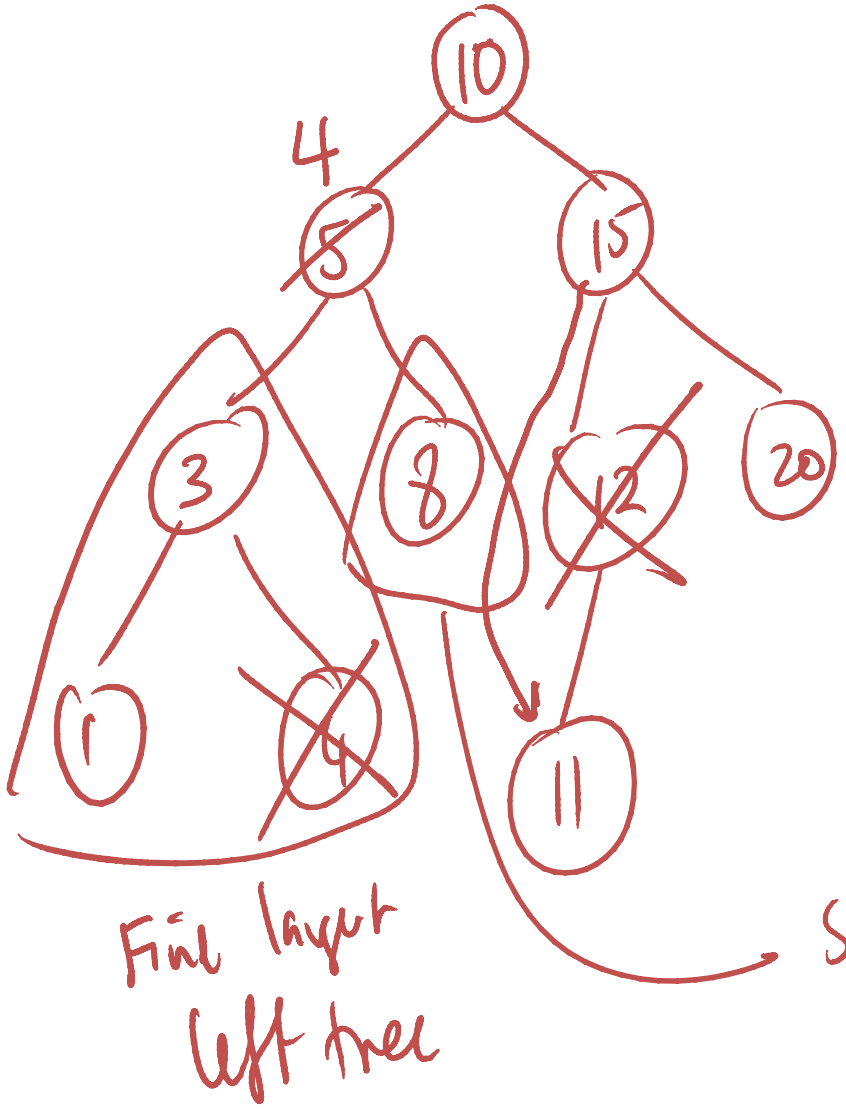


Full
Complete
Perfect

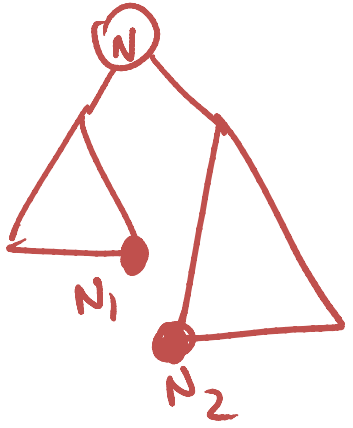
insert
delete

delete(11) → easy → (12).left = null

delete(12) → (15).left = (12).left



delete(5) →



$N_1 < N < N_2$