COP 3502 2/17/26

1) Final Project
2) Recurrence Relations
3) Linked Lists

```
modpow(~, int exp) {
    if (exp == 0) return 1;
    int tmp = modpow(~, exp/2);
    return (tmp*tmp)%mod;
}
```

O(1) time

$T(\frac{e}{2})$

Let e = value exponent

Let T(e) = run time for exponent e

$$T(e) = T(\tfrac{e}{2}) + O(1)$$

$T(\frac{e}{2}) = T(\frac{e}{4}) + O(1)$

$T(\frac{e}{4}) = T(\frac{e}{8}) + O(1)$

$$= T(\tfrac{e}{4}) + O(1) + O(1) \quad \text{1st iter}$$

$$= T(\tfrac{e}{4}) + 2O(1) \quad \text{2nd iter}$$

$$= T(\tfrac{e}{8}) + O(1) + 2O(1)$$

$$= T(\tfrac{e}{8}) + 3O(1) \quad \text{3rd iter}$$

After k iterations $= T(\frac{e}{2^k}) + kO(1)$

We know $T(1) = 1$

For what value of k does $\frac{e}{2^k} = 1$

$e = 2^k \leftrightarrow k = \log_2 e$, Plug in:

$$T(n) = T\left(\frac{e}{2^{\log_2 e}}\right) + \log_2 e \cdot O(1)$$

$$= T(1) + O(\log e)$$

$$= O(1) + O(\log e)$$

$$= \boxed{O(\log e)}$$

MS(n)

MS(n/2) ✓

MS(n/2) ✓

for loop runs n iterations

} Simplification of Merge Sort

Let $T(n)$ = run-time of MS of n items

$$T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{2}\right) + O(n)$$

$$T(n) = \boxed{2T\left(\frac{n}{2}\right) + n}, \quad T(1) = 1$$

$$= 2\left[2T\left(\frac{n}{4}\right) + \frac{n}{2}\right] + n$$

$$= 4T\left(\frac{n}{4}\right) + n + n$$

$$= \boxed{4T\left(\frac{n}{4}\right) + 2n}$$

$$= 4\left(2T\left(\frac{n}{8}\right) + \frac{n}{4}\right) + 2n$$

$$= \boxed{8T\left(\frac{n}{8}\right) + 3n}$$

$T\left(\frac{n}{2}\right) = 2T\left(\frac{n}{4}\right) + \frac{n}{2}$

$T\left(\frac{n}{4}\right) = 2T\left(\frac{n}{8}\right) + \frac{n}{4}$

After k iter: $T(n) = 2^k T\left(\frac{n}{2^k}\right) + kn$

let $\frac{n}{2^k} = 1$, $2^k = n$ $k = \log_2 n$ plug in:

$$T(n) = nT(1) + (\log_2 n)\, n$$

$$T(n) = n + O(n \lg n)$$

$$= O(n \lg n)$$

Master Theorem

$$T(n) = AT\left(\frac{n}{B}\right) + O(n^k)$$

$$\begin{cases} \text{if } B^k < A, & O(n^{\log_B A}) \\ \text{if } B^k = A, & O(n^{\log_B A} \cdot \lg n) \\ \text{if } B^k > A, & O(n^k) \end{cases}$$

1) $A = 2, B = 2, k = 1, \quad 2^1 = 2$ ✓ Case 2 $O(n^{\log_2 2} \cdot \lg n)$

$T(n) = 3T(n-1) + 3^n, \quad n > 1$

$T(1) = 12$

$T(n) = \boxed{3T(n-1) + 3^n}$

$= 3(3T(n-2) + 3^{n-1}) + 3^n$

$= 9T(n-2) + 3^n + 3^n$

$= \boxed{9T(n-2) + 2 \times 3^n}$

$= 9(3T(n-3) + 3^{n-2}) + 2 \times 3^n$

$= 27T(n-3) + 3^n + 2 \times 3^n$

$= \boxed{27T(n-3) + 3 \times 3^n}$

$T(n-1) = 3T(n-2) + 3^{n-1}$

$T(n-2) = 3T(n-3) + 3^{n-2}$

$9 \times 3^{n-2} = 3^2 \times 3^{n-2} = 3^{2+n-2} = 3^n$

After k iter $T(n) = 3^k T(n-k) + k3^n$

Let $n-k = 1$, $k = n-1$ $\quad T(1) = 12$

$T(n) = 3^{n-1} T(1) + (n-1)3^n$

$= 12 \times 3^{n-1} + (n-1)3^n$

$= 4 \times 3 \times 3^{n-1} + (n-1)3^n$

$= 4 \times 3^n + (n-1)3^n$

$= \boxed{3^n(n+3)}$

$T(n) = (n+1)T(n-1)$, for $n > 1$

$T(1) = 1$

$T(n-1) = nT(n-2)$

$T(n-2) = (n-1)T(n-3)$

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

$= (n+1) \cdot n \cdot T(n-2)$

$= (n+1) \cdot n \cdot (n-1) \cdot T(n-3)$

After k iter $T(n) = (n+1)(n)(n-1)(n-k+2)T(n-k)$

Let $n-k = 1$, so $k = n-1$

$T(n) = (n+1)(n)(n-1) \cdots (n-(n-1)+2)T(1)$

$T(n) = (n+1)(n)(n-1) \cdots (3)T(1)$

$T(n) = \frac{(n+1)(n)(n-1) \cdots (3)(2)(1)}{(2)(1)}$

$= \boxed{\frac{(n+1)!}{2}}$

Linked Lists

NULL

1) create node

2) ~~ins~~ insert in order (insert to front)

3) print

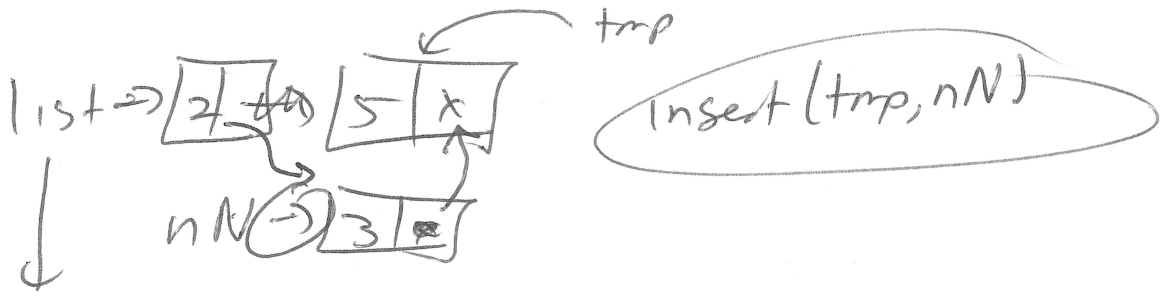
```
typedef struct node {
    int data;
    struct node* next;
} node;
```

---

main

list = insertInOrder (list, nN)

[if list is NULL
we'll return nN.
if nN goes first, DO
nN->next = list;
return nN;



list->next = insert