Complexity anlaysis:

Repeated squaring:

T(n) = T(n/2) + T(n/2) + C twice

= 2T(n/2) + C

= 2(2T(n/4) + C) +C

= 4T(n/4) + 2C + C

…

= nT(1) + C + 2C + 4C +...+2xC

~n(T(1) + C) O(n)

pow(a, n/2)\*p(a, n/2)

Tmp = pow(a, n/2)

tmp\*tmp

T(n) = T(n/2) + K (K>C)

= T(n/4) + K + K

…

= T(1) + log(n)\*k

~klog(n) = O(log n)

Dynamic Programming (DP)

Save result

“Master theorem”

fib(n) = fib(n-1) + fib(n-2)

T(n) = T(n-1)+T(n-2)

To score credit:

Boundary condition:

Def fib(n):

If n < 0:

print(‘error’)

return

If n == 0 || n == 1:

Return n

Return fib(n-1)+fib(n-2)

DP:Use additional space to change an otherwise infeasible problem to computable

Trade space for time:

fib(n): n = 0 to n

PL index from 1; 0

fib(0)= 0

fib(1) = 1

fib(2) = 1

fib(3) = 2

…

fib(n) =

Prev2 = 0

Prev1 = 1

For ix in range (2, n):

New\_v = prev1 +prev2

No free lunch

Exponential: infeasible

Alg efficiency

T(n) = an + C

T(n) = T(n-1) + T(n-2)

an = a(n-1)+a(n-2)^

Div a(n-2):

a2 + a + 1

a2 - a - 1 = 0

Solve: a = 1.618 golden ratio

fib(n) O(1.618n)

n = 1M, 1.6181000

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

=... = nT(1) + n log n

O() //couldn’t get the rest of this

O(1)

O(n)

fib(n) = c1 an+ c2 a2n

a1, a2, c1, c2 are given constant

O(log n): repeated squaring

fib(n):1.618n -> n -> O(log n)

Not computable -> most efficient

a2 - a - 1 = 0

reasonable/educated guess of T(n) (Order of T):

T(n) is O(nk)

= T(n-2)+T(n-3)+T(n-2)

----------------------

Agile: standup, sprint, PI-meeting (Principal Investigator/leader)

Weekly, bi-weekly, quarter //(respectively)

Shorter (0.5 hr), 2-3 hrs, PI: whole day 9-4

Analyze system: all perspectives are exhausted

Blind people: elephant touch micro-region

Tree trunk: leg

Rope: nose

Fan: ear

Knife: tusk

…

Perspective-based method:

UML: unified modeling language

SE Industry standard to describe a system

Use case diagram: Simplest UML diagram

Use ovals/ellipse to rep system features

Find out all possible users of the system

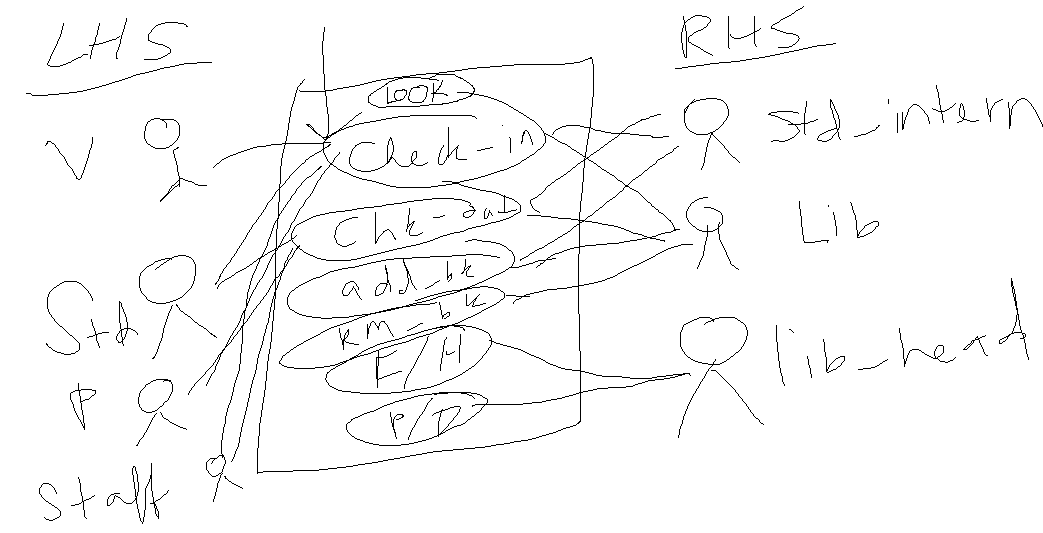
Link users to diff ovals

CCNY lib MIS:

ID all users: std (students); prof; libr (librarians); std\_interns; staff; lib\_head; visitors

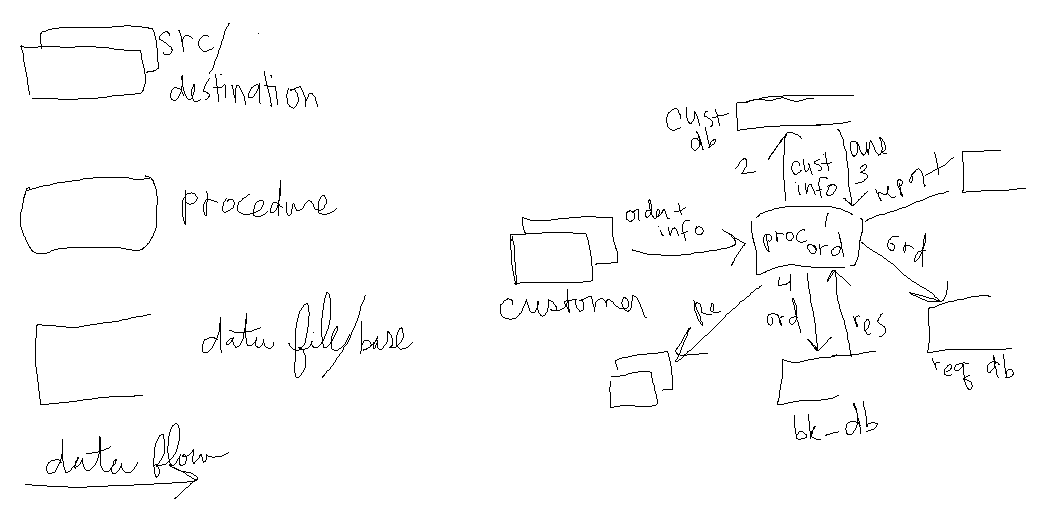
ID features: chk\_out; chk\_in; add\_bk; add\_bk; rm\_bk;

hire/fire lib; prom/dem lib; punish\_user



Data-flow diagram (DFD):

Used to be the kind of all SE diagrams (but no longer)



New King: Class Diagram (similar)