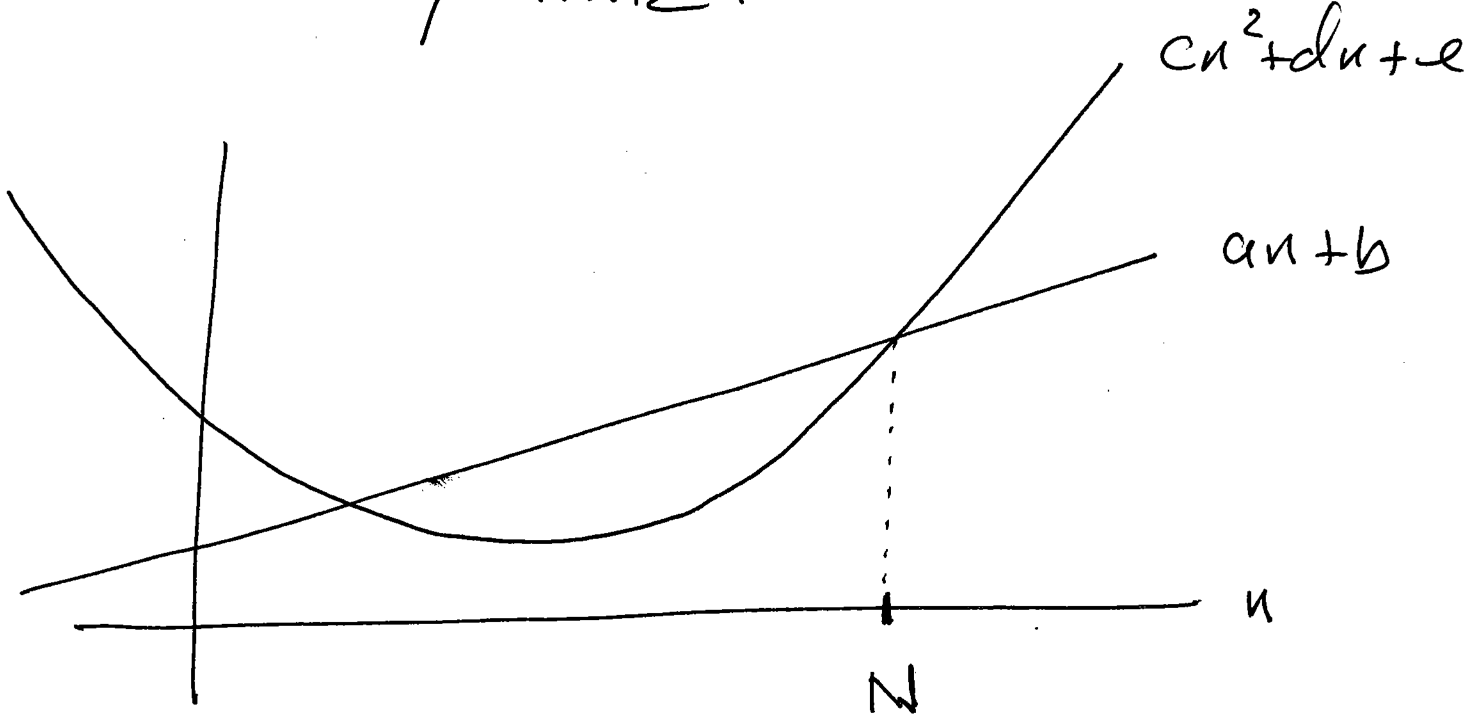
CMPS 10 2-5-08

RECALL: EVERY PARABOLA IS EVENTUALLY ABOVE EVERY LINE:



<u>THM</u> GIVEN FUNCTIONS $an+b$ AND cn^2+dn+e (WHERE $a>0$, $c>0$) THERE EXISTS AN $N>0$ S.T. FOR ALL $n>N$:

$$cn^2+dn+e > an+b$$

HOLDS.

<u>TERMINOLOGY</u>: WE SAY $an+b$ IS <u>OF ORDER</u> n WRITE $an+b = \Theta(n)$. WE SAY cn^2+dn+e IS <u>OF ORDER</u> n^2, AND WRITE $cn^2+dn+e = \Theta(n^2)$

WHAT IS MEANT BY $\Theta(g(n))$ i.e. 'ORDER $g(n)$'

FORMAL DEFN

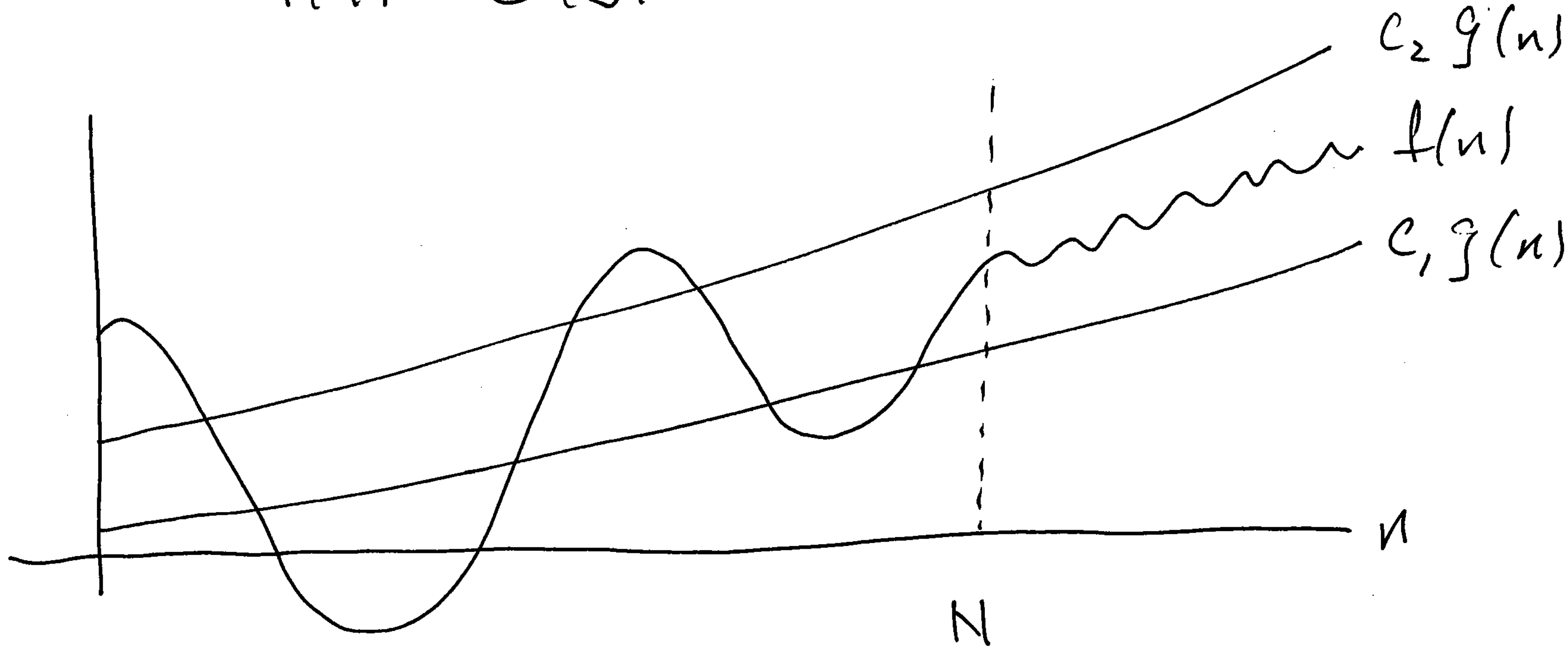
A FUNCTION $f(n)$ IS SAID TO BE **OF ORDER $g(n)$**

IFF THERE EXIST POS. CONSTANTS c_1, c_2, AND N

s.t. FOR ALL $n > N$:

$$0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$$

NOTATION: $f(n) = \Theta(g(n))$



INFORMAL DEFN:

$f(n) = \Theta(n^2)$ <u>if</u> $f(n) = \text{const} \cdot n^2 + L.O.T.$ (lower order terms)

(L.O.T. = TERMS THAT GROW SLOWER THAN n^2.)

i.e. $f(n) = \Theta(n^2)$ <u>if</u> $\frac{f(n)}{n^2} = \text{const} + \frac{L.O.T.}{n^2}$

($\frac{L.O.T.}{n^2} \to 0$ AS $n \to \infty$)

i.e. $f(n) = \Theta(n^2)$ <u>if</u> $\frac{f(n)}{n^2} \to \text{const}$ AS $n \to \infty$

$f(n) = \Theta(g(n))$ <u>if</u> $f(n) = \text{const} \cdot g(n) + \text{L.O.T.}$

L.O.T. = TERMS THAT GROW SLOWER THAN $g(n)$

i.e. <u>if</u> $\dfrac{f(n)}{g(n)} = \text{const} + \dfrac{\text{LOT}}{g(n)}$, where $\dfrac{\text{LOT}}{g(n)} \to 0$ AS $n \to \infty$

i.e. <u>if</u> $\dfrac{f(n)}{g(n)} \to \text{const}$ AS $n \to \infty$

Ex. $f(n) = 12n^3 - 5n^2 + 14n + 3$

FACT: $f(n) = \Theta(n^3)$

CHECK: $\dfrac{f(n)}{n^3} = 12 - \dfrac{5}{n} + \dfrac{14}{n^2} + \dfrac{3}{n^3}$

$\dfrac{5}{n} \to 0$, $\dfrac{14}{n^2} \to 0$, $\dfrac{3}{n^3} \to 0$ AS $n \to \infty$

SO $\dfrac{f(n)}{n^3} \to 12$ AS $n \to \infty$

EX. SUPPOSE WE HAVE A PC & A SUPER COMPUTER (SC). PC DOES 100 MEGA FLOPS (100 million FLOATING POINT OPS PER SEC.) SC DOES 500 GIGA FLOPS (500 Billion FLOPS).

$$PC: \quad 10^{8} \frac{OP}{SEC.} \quad \approx \quad 10^{-8} \frac{SEC}{OP}$$

$$SC: \quad 5 \cdot 10^{11} \frac{OP}{SEC} \quad \approx \quad 2 \cdot 10^{-12} \frac{SEC}{OP.}$$

SUPPOSE PC RUNS AN ALGORITHM WITH $\Theta(n)$ RUNTIME, AND SC RUNS AN ALGORITHM WITH $\Theta(n^2)$ RUNTIME.

INPUT SIZE n	PC : n TIME	SC : n^2 TIME
10^2	10^{-6} SEC.	$2 \cdot 10^{-8}$ SEC
10^3	10^{-5}	$2 \cdot 10^{-6}$ SEC
10^4	10^{-4}	$2 \cdot 10^{-4}$
10^5	10^{-3}	$2 \cdot 10^{-2}$
10^6	10^{-2}	2 SEC
10^7	10^{-1}	200 SEC
10^8	1 SEC	20000 SEC = 5.5 HRS
10^9	10 SEC.	$2 \cdot 10^6$ SEC = 550 HRS.

Binary Search

Ex. $n = 10$, target 11

list:

| 2 3 4 7 (10) 11 15 20 25 28 |

L ... R

2 3 4 7 10 | 11 15 (20) 25 28 |

L ... R

2 3 4 7 10 | (11) 15 | 20 25 28

L ... R

# of comp = 3

Ex. Same list, target = 3

| 2 3 4 7 (10) 11 15 20 25 28 |

L ... R

| 2 (3) 4 7 |

L ... R

\# comp = 2

Ex target = 7

| 2 3 4 7 (10) 11 15 20 25 28 |

L R

| 2 (3) 4 7 |

L R

| (4) 7 |

L R

#comp = 4

| (7) |

L R

Ex. target = 8

← 5.5

1	2	3	4	5	6	7	8	9	10
2	3	4	7	(10)	11	15	20	25	28

L ... R

| 2 (3) 4 7 |

L ... R

#comp = 4

| (4) 7 |

L ... R

| (7) |

L R

| |

R L

Search space is now empty.

<u>INPUT</u> : $n \geq 1$: # OF #S IN list

$a_1, \ldots, a_n$: the list (SORTED !!)

target : to search for

<u>OUTPUT</u> : THE INDEX m s.t. $a_m = \text{target}$

OR 0 IF NO SUCH m EXISTS.

(ASSUME $a_1, \ldots, a_n$ DISTINCT)

Binary Search

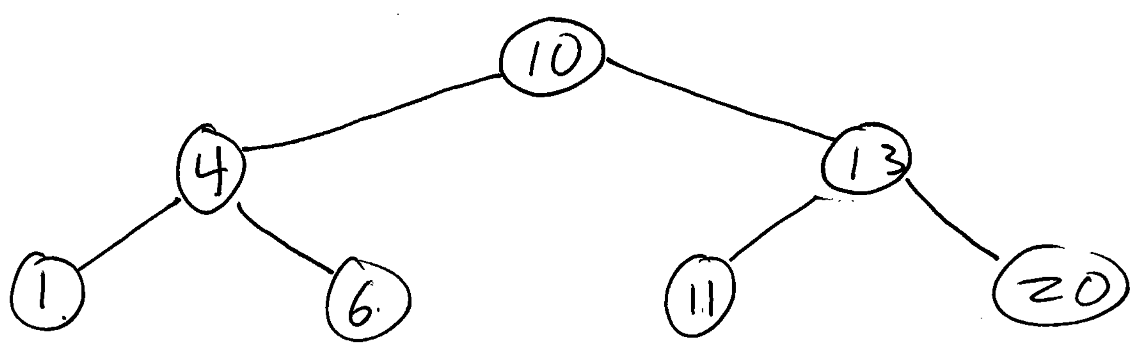
```
L ← 1
R ← n
found ← false
while L ≤ R and not found
    m ← ⌊(L+R)/2⌋
    if target = a_m
        found ← true
    else if target < a_m
        R ← m-1
    else
        L ← m+1
if not found
    m ← 0
Print m
```

NOTE: $\lfloor x \rfloor$ DENOTES THE FLOOR OR INTEGER PART OF x.

$\lfloor 6 \rfloor = 6$, $\lfloor 6.5 \rfloor = 6$.

BINARY SEARCH TREE

EX. 1 4 6 10 11 13 20



#COMPS

1

2

3