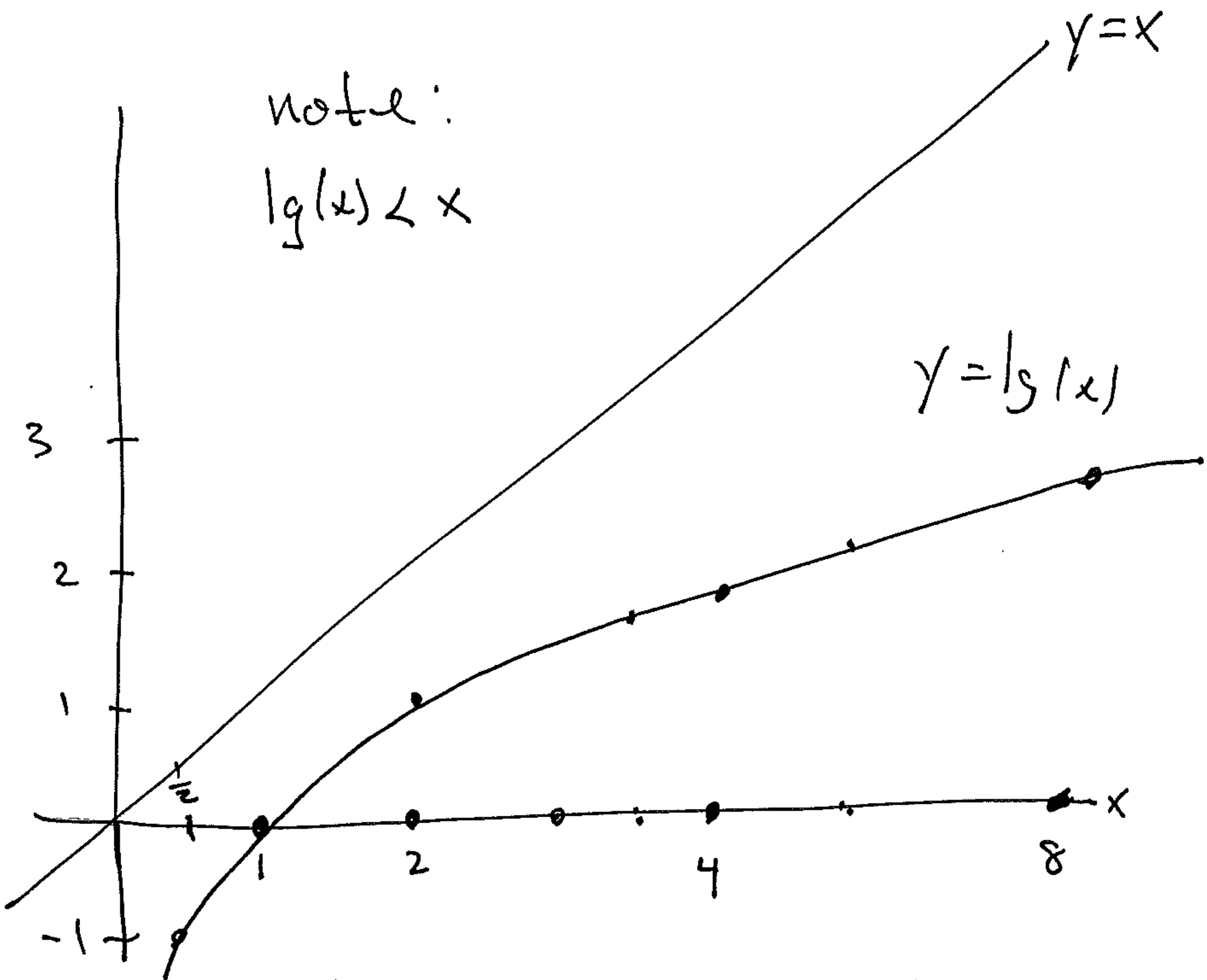
CMPS 10 10-29-09

# Graph of $y = \lg(x)$




$\lg(1) = 0$ $\lg(2) = 1$ $\lg(4) = 2$ $\lg(8) = 3$

$2^0 = 1$ $2^1 = 2$ $2^2 = 4$ $2^3 = 8$

$\lg(\frac{1}{2}) = -1$

Observe:

lg(n) grows very slowly with n

Any Algorithm with lg(n) Run-Time is very efficient.

Recall:

$$\boxed{2^{k-1}-1 < n \leq 2^{k}-1 \iff W(n)=k}$$

$$\therefore \quad 2^{k-1} < n+1 \leq 2^{k}$$

$$\therefore \quad 2^{k-1} \leq n < 2^{k}$$

$$\therefore \quad \lg(2^{k-1}) \leq \lg(n) < \lg(2^{k})$$

$$\therefore \quad k-1 \leq \lg(n) < k$$

$$\therefore \quad k-1 = \lfloor \lg(n) \rfloor$$

$\therefore \quad K = \lfloor \lg n \rfloor + 1$

So we've proved

$$\boxed{W(n) = \lfloor \lg n \rfloor + 1}$$

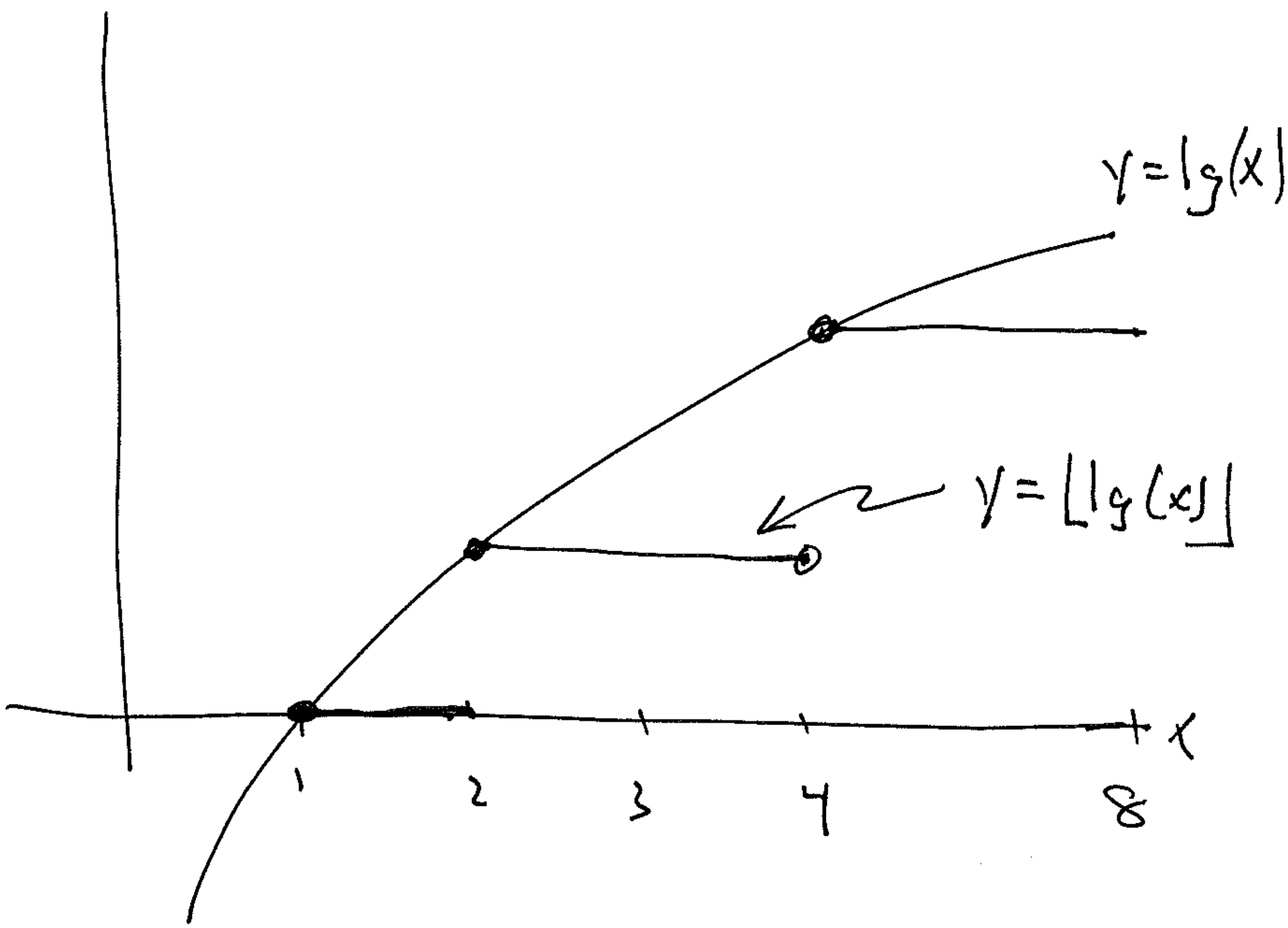
<u>Ex.</u> $W(100) = \lfloor \lg(100) \rfloor + 1$

$= \lfloor 6.??? \rfloor + 1$

$= 6 + 1 = 7$

Fact: $\lfloor \lg n \rfloor = \Theta(\lg n)$

$$\therefore \quad \boxed{W(n) = \Theta(\lg(n))}$$



y = lg(x)
y = ⌊lg(x)⌋
x
1
2
3
4
8

Problem:

Given an unsorted list

$a_1, \ldots, a_n$, to be searched,

shoulo we

cost

(1) use seq. search : $\Theta(n)$

or

(2) first sort, then use binary search.

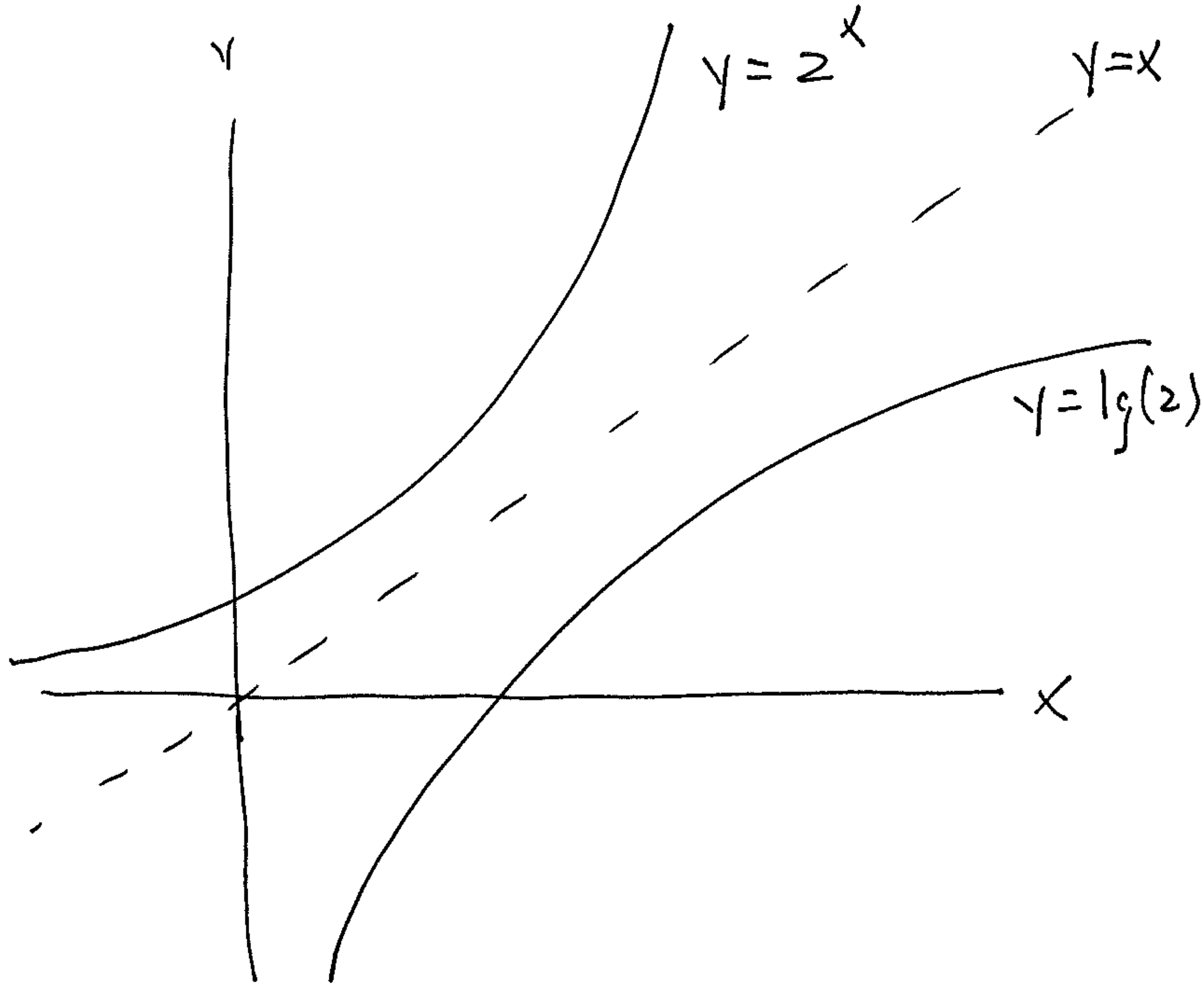
$$\underbrace{\text{assume cost} = \Theta(n^2) \qquad \Theta(\lg n)}$$

total cost $= \Theta(n^2 + \lg n) = \Theta(n^2)$

Answer: If you will search for just one target use (1). If many targets, sort once, then search with Binary Search.

Defn

An Algorithm is said to be of Exponential order, if its runtime is $\Theta(2^n)$ (or $\Theta(b^n)$ for any $b > 1$.)



Any exponential Algorithm is practical for only small values of n.

Examples:

- Brute force Chess analysis,
- Traveling Salesman
- Hamiltonian Circuits

# Chapter 4: logic design/circuits

- Base b Positional numbering system (for $b > 1$).

Base 10:

what does 12526 mean?

$$[12526]_{10} = 1 \cdot 10^4 + 2 \cdot 10^3 + 5 \cdot 10^2 + 2 \cdot 10^1 + 6 \cdot 10^0$$

what is Base b?

Given $b > 1$ (an integer), assign b symbols to the integers:

$$0, 1, 2, \ldots, b-1$$

call these symbols Digits

A string of $n$ digits

$$\left[a_{n-1}a_{n-2}\cdots a_2 a_1 a_0\right]_b$$

then stands for the integer:

$$a_{n-1}\cdot b^{n-1} + a_{n-2}\cdot b^{n-2} + \cdots + a_2\cdot b^2 + a_1\cdot b^1 + a_0\cdot b^0$$

more generally a fraction is Represented as:

$$\left[a_{n-1}\cdots a_0 . a_{-1}\cdots a_{-k}\right]_b$$

$$= a_{n-1}\cdot b^{n-1} + \cdots + a_0\cdot b^0 + a_{-1}\cdot b^{-1} + \cdots + a_{-k}\cdot b^{-k}$$

# Important bases in C.S.

$b=2$ : Binary. Digits $= \{0, 1\}$

$b=8$ : Octal. $\{0, 1, 2, 3, 4, 5, 6, 7\}$

$b=16$ : Hexidecimal

$$\{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, \underset{10}{A}, \underset{11}{B}, \underset{12}{C}, \underset{13}{D}, \underset{14}{E}, \underset{15}{F}\}$$

$b=10$ : Decimal $\{0, 1, \ldots, 9\}$

# Ex.

$$[1011010111001]_2$$

$$= 1\cdot 2^{12} + \cancel{0\cdot 2^{11}} + 1\cdot 2^{10} + 1\cdot 2^{9} + \cancel{0\cdot 2^{8}} + 1\cdot 2^{7} + \cancel{0\cdot 2^{6}}$$

$$+ 1\cdot 2^{5} + 1\cdot 2^{4} + 1\cdot 2^{3} + \cancel{0\cdot 2^{2}} + \cancel{0\cdot 2^{1}} + 1\cdot 2^{0}$$

$$= 4096 + 1024 + 512 + 128 + 32 + 16$$

$$+ 8 + 1 = [5817]_{10}$$

$1 = 2^0$

$2 = 2^1$

4

8

16

32

64

128

256

512

1024

2048

$4096 = 2^{12}$

Ex. $[237]_8$

$= 2 \cdot 8^2 + 3 \cdot 8^1 + 7 \cdot 8^0$

$= 2 \cdot 64 + 3 \cdot 8 + 7$

$= 128 + 24 + 7 = 159$

Ex. $[A17D]_{16}$

$= A \cdot 16^3 + 1 \cdot 16^2 + 7 \cdot 16^1 + D \cdot 16^0$

$= 10 \cdot 16^3 + 16^2 + 7 \cdot 16 + 13 \cdot 1$

$= 41\,341$