CMPS 10 1-31-08

Insertion Sort

Ex 2 | 4 6 1 5 3 (n = 6)

2 4 | 6 1 5 3

2 4 6 | 1 5 3

2 4 1 6 5 3

2 1 4 6 5 3

1 2 4 6 | 5 3

1 2 4 5 6 | 3

1 2 4 5 3 6

1 2 4 3 5 6

1 2 3 4 5 6

INPUT: $n, a_1, \ldots, a_n$

OUTPUT: MODIFIED LIST IN INCREASING ORDER

Insertion Sort

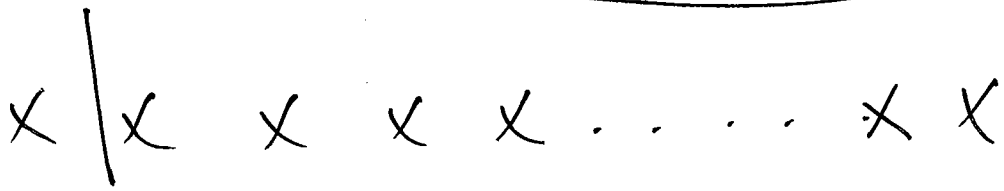
```
1.) L ← 2
2.) while L ≤ n
3.)     j ← L
4.)     while j ≥ 2 and a_j < a_{j-1}
5.)         swap a_j ↔ a_{j-1}
6.)         j ← j - 1
7.)     L ← L + 1
8.) stop
```

Run Time Analysis

Basic op is comparison of list elements.

Best case: occurs when list is already sorted.

$\#\text{comp} = n-1$ ✓



WORST CASE: occurs when list is sorted in DECREASING ORDER.

$n = 6$

	# comp
6 | 5 4 3 2 1	1
5 6 | 4 3 2 1	2
4 5 6 | 3 2 1	3
3 4 5 6 | 2 1	4
2 3 4 5 6 | 1	5
1 2 3 4 5 6 |	15

$$1+2+3+4+5 = \frac{5 \cdot 6}{2} = 15$$

IN GENERAL (i.e. FOR ARBITRARY n) INSERTION SORT DOES IN WORST CASE:

$$\#\text{comp} = \frac{n(n-1)}{2}$$

i.e. $\#\text{comp} = 1 + 2 + 3 + \cdots + (n-1) = \frac{(n-1)(n-1+1)}{2}$

RECALL $1 + 2 + 3 + \cdots + \text{☺} = \frac{\text{☺}(\text{☺}+1)}{2}$

i.e. $\#\text{comp} = \frac{1}{2}n^2 - \frac{1}{2}n$

Ex. Suppose we have 2 Algorithms A & B which solve the same problem.

Suppose they do

A : $1000n$ basic ops.

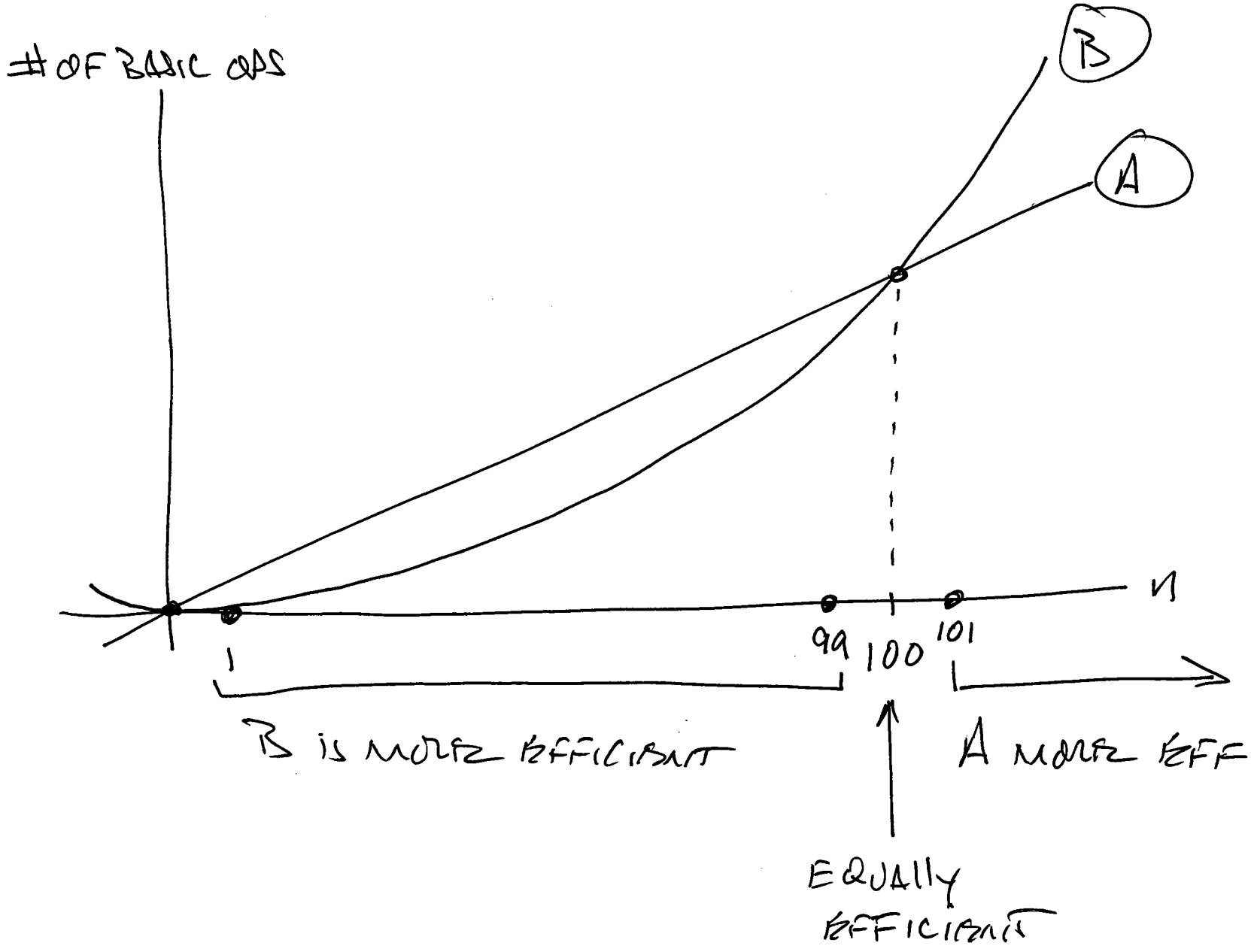
B : $10n^2$ basic ops.

On input of size n (say in worst case)

Which one should be considered more efficient?

solve : $1000n = 10n^2$

$100 = n$



A is more eff. for larger values of n.

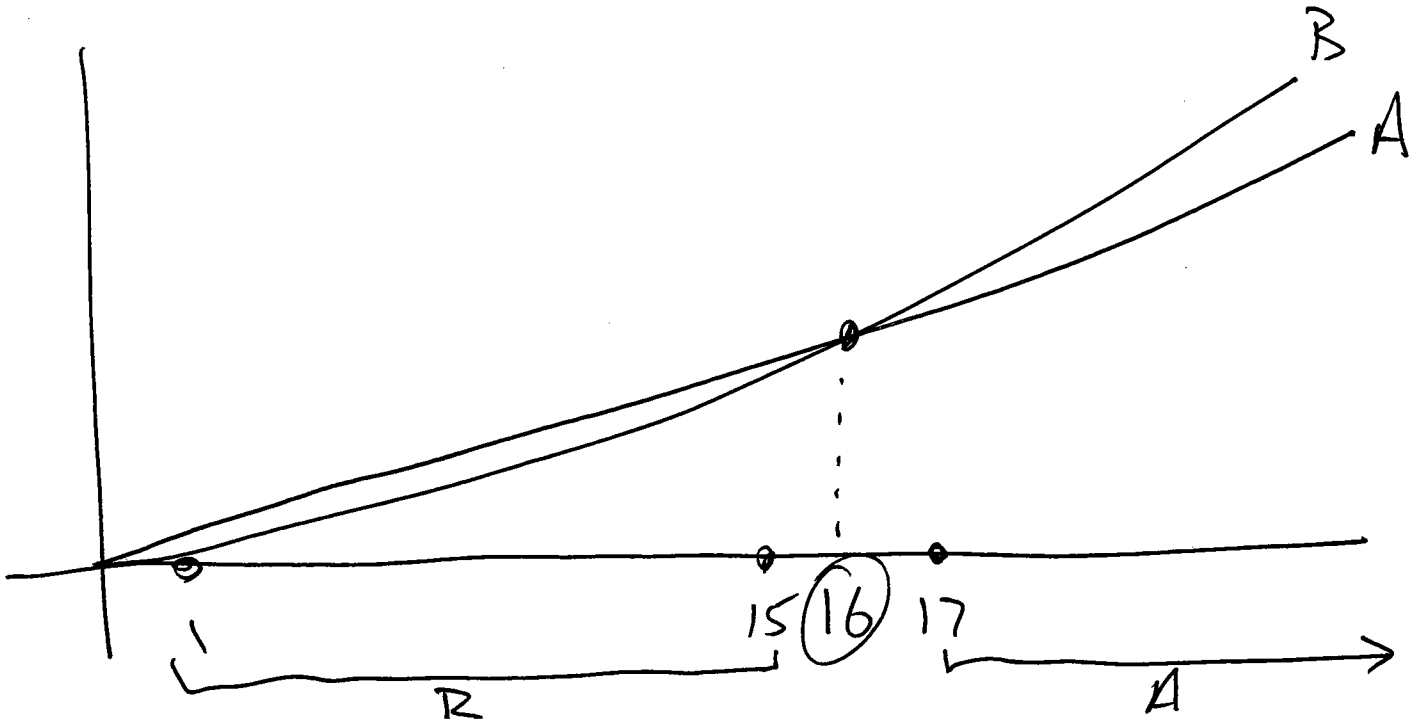
EX. SAME QUESTION

A: $12n\sqrt{n} = 12n^{3/2}$ BASIC OPS.

B: $3n^2$ " "

$12n\sqrt{n} = 3n^2 \Rightarrow 4\sqrt{n} = n = \sqrt{n}\cdot\sqrt{n}$

$\Rightarrow 4 = \sqrt{n} \Rightarrow n = 16$



IN GENERAL, IF

A: an BASIC OPS

B: bn^2 " "

THEN A IS MORE EFF. FOR LARGE n,

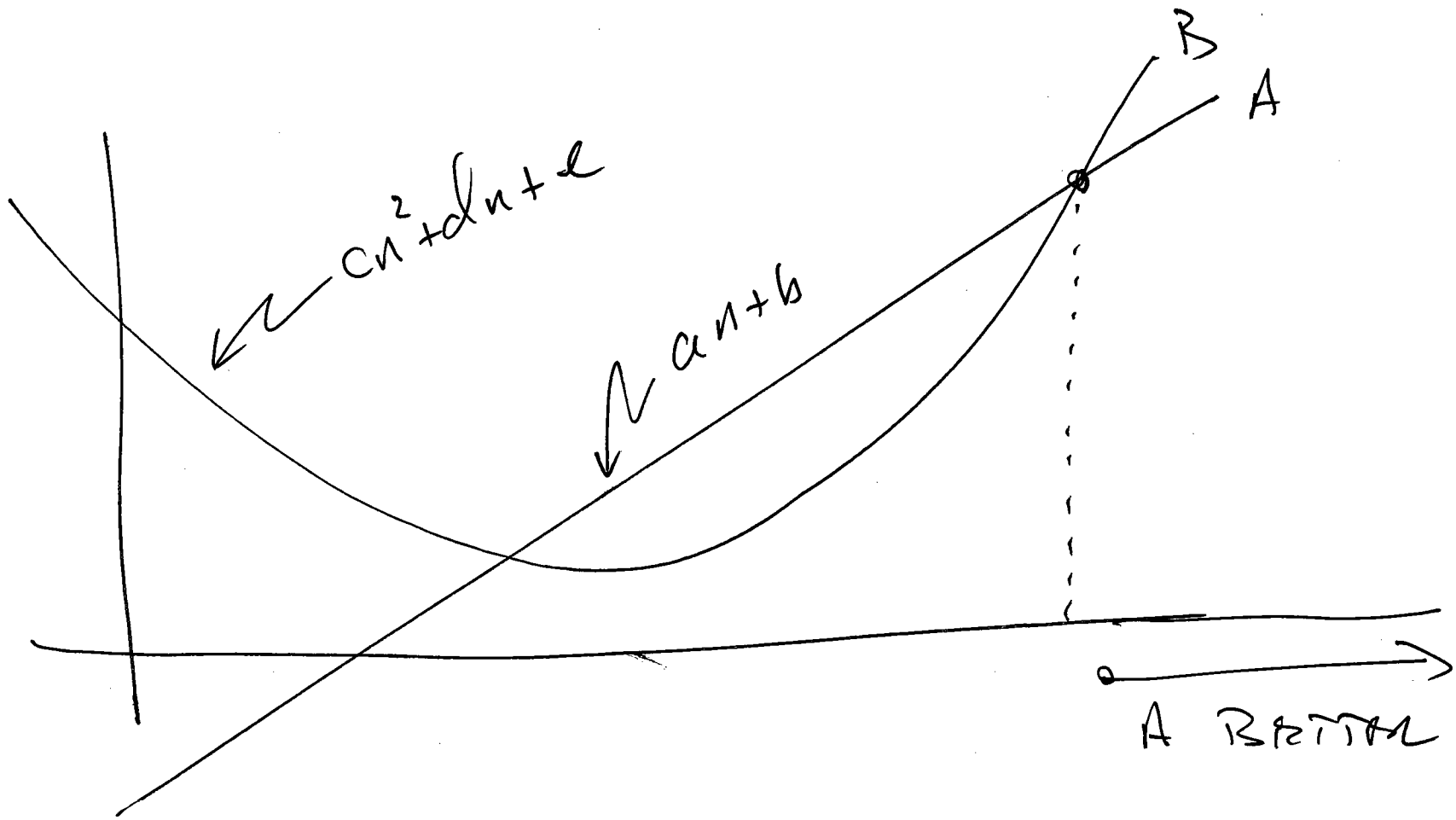
NO MATTER WHAT a & b are!

NOTE: ANY PARABOLA IS EVENTUALLY ABOVE

ANY LINE.

THIS IS TRUE EVEN WITH LOWER ORDER

TERMS PRESENT



WE CLASSIFY ALL ALGORITHMS WITH RUN TIME $an+b$ TO BE IN CLASS $\Theta(n)$ (ORDER n)

LIKEWISE

cn^2+dn+e ..... $\Theta(n^2)$ (ORDER n^2)