CMPS 10 1-24-08

EFFICIENCY OF ALGORITHMS

1.) MUST SEEK A NUMBER OF RUN TIME WHICH IS MACHINE INDEPENDENT

2.) MUST TAKE INTO ACCOUNT ALL POSSIBLE INPUTS OF A GIVEN SIZE.

TO DEAL WITH (1) WE COUNT # OF INSTRUCTIONS EXECUTED INSTEAD OF # OF SECONDS.

EX. NOT ALL INSTRUCTIONS SHOULD BE COUNTED EQUALLY

```
:
if a < b
    a ← b
else
    print c
:
```

PROCEDURE: PICK SOME BASIC OPERATION & COUNT # OF TIMES IT IS EXECUTED.

TO DEAL WITH (2) WE CONSIDER 3 DIFFERENT MEASURES

- WORST CASE: MAXIMUM COST TAKEN OVER ALL INPUTS OF A GIVEN SIZE

- **Best case**: minimum cost taken over all inputs of a given size
- **Average case**: average cost over all possible inputs of fixed size.

Recall: Sequential search.

Input: $n \geq 1$, $a_1, \ldots, a_n$, target

Output: 1st index i s.t. $a_i = \text{target}$, or 0 if no such i exists.

SEQUENTIAL SEARCH

```
1.) i ← 1
2) found ← false
3.) while i ≤ n and not found
4.)     if a_i = target
5.)         found ← true
6.)     else
7.)         i ← i+1
8.) if not found
9.)     i ← 0
10.) Print i
11.) stop
```

BASIC OPERATION: COMPARISON OF target TO A # IN THE LIST, i.e. WE COUNT # OF TIMES LINE 4 is EXECUTED.

NOTE: • LINES 6 & 11 DON'T REALLY DO ANYTHING.

- LINES 1, 2, 8, 10 ARE EXECUTED ONCE.
- LINES 5, 9 is EXECUTED AT MOST ONCE.
- LINES 3, , 7 ARE EXECUTED (APPROXIMATELY) THE SAME # OF TIMES AS LINE 4.

BEST CASE OBVIOUSLY OCCURS WHEN a_1 = target

WHENCE # OF COMPARISONS = 1

WORST CASE OCCURS WHEN target = a_n

(AND target $\neq a_k$ FOR ANY $k < n$). IN THIS

CASE #COMPARISONS = n.

AVERAGE CASE: ASSUME target IS IN list

AND IS EQUALLY LIKELY TO BE AT ANY

POSITION IN THE list. ALSO ASSUME

$a_1, \ldots, a_n$ ARE DISTINCT.

if target is in Pos 1 : # comp = 1

" " " " 2 : # comp = 2

" " " " 3 : " = 3

⋮

" " " " n : # comp = n

So AVERAGE # COMPARISONS is

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

NOTICE: IF WE HAD COUNTED PERIPHERAL OPERATIONS OUR ANSWERS WOULD BE:

		WE GOT
WORST CASE:	$an + b$	n
AVG. CASE:	$cn + d$	$\frac{1}{2}n + \frac{1}{2}$
BEST CASE:	e	1

WHERE $a - e$ ARE CONSTANTS THAT DEPEND ON ACTUAL COMPUTING DEVICE

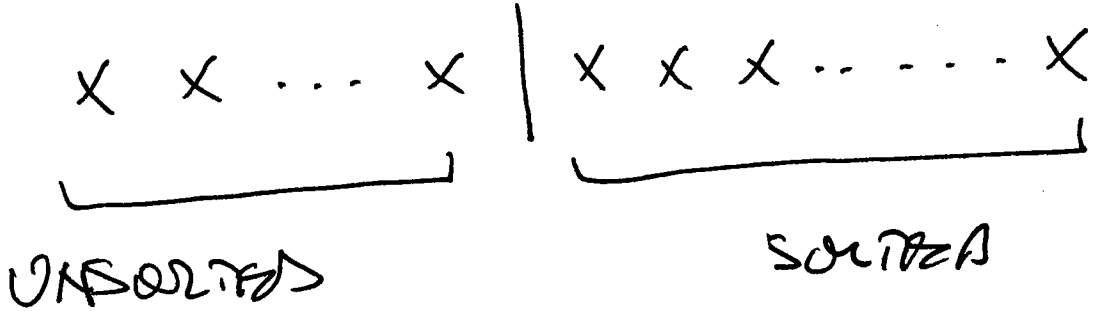
NEW PROBLEM: SORTING.

INPUT: $n \geq 1$ (# OF #S IN LIST)

$a_1, \ldots, a_n$ (LIST OF #S TO BE SORTED)

OUTPUT: A RE-ARRANGEMENT OF INPUT LIST IN WHICH $a_1 \leq a_2 \leq a_3 \leq \ldots \leq a_n$

SELECTION SORT:



Ex. $n = 6$