LEC 7 09/10/99

ALGORITHMS AND COMPLEXITY

ALGORITHM — A FINITE SETS OF PRECISE INSTRUCTIONS FOR PERFORMING A COMPUTATION

EXAMPLE: FINDING THE MAXIMUM ELEMENT

```
PROCEDURE max (a_1, a_2, ..., a_n : INTEGERS)
  max := a_1
FOR i=2 TO n
    IF max < a_i THEN max := a_i
{max IS THE LARGEST ELEMENT}
```

↑ ↑ ↑ ↑ ↑ ↑

WRITTEN IN PSEUDOCODE

$\underbrace{\text{INPUT / OUTPUT, DEFINITENESS, FINITENESS}}_{\text{BASIC}}$ $\qquad$ $\underbrace{\text{CORRECTNESS, GENERALITY, EFFECTIVENESS}}_{\text{DESIRED}}$

PROPERTIES: BASIC / DESIRED

NUMBER OF COMPARISONS IN max $= 2(n-1)+1 = 2n-1$. Complexity $O(n)$.

THE LINEAR SEARCH ALGORITHM

PROCEDURE LINEAR SEARCH (x: INTEGER, $a_1, a_2, \ldots, a_n$: DISTINCT INTEGERS)
$i := 1$
WHILE ($i \le n$ AND $x \ne a_i$)
    $i := i + 1$
IF $i \le n$ THEN *location* $:= i$
ELSE *location* $:= 0$
{ *location* IS THE INDEX OF TERM THAT EQUALS x, OR 0 IF x IS NOT FOUND }

GOAL: LOCATE AN ELEMENT x IN THE LIST OF DISTINCT ELEMENTS $a_1, a_2, \ldots, a_n$

NUMBER OF COMPARISONS C:
IF X IS A_1 $\quad C = 3$
IF X IS A_i $\quad C = 2i + 1$
IF X IS NOT IN THE LIST $\underbrace{C = 2n + 2}_{\text{WORST CASE}}$

COMPLEXITY IS $O(n)$

BINARY SEARCH ALGORITHM

```
PROCEDURE binary search (x: INTEGER,
a_1, a_2, ..., a_n : INCREASING INTEGERS)
i := 1 {i IS LEFT ENDPOINT OF SEARCH INTERVAL}
j := n {j IS RIGHT ENDPOINT OF SEARCH INTERVAL}
WHILE i < j
BEGIN
    m := ⌊(i+j)/2⌋
    IF x > a_m THEN i := m+1
    ELSE j := m
END
IF x = a_i THEN location := i
ELSE location := 0
```

ASSUME FOR SIMPLICITY $n = 2^k$ $(k = \log n)$

TWO COMPARISONS AT EACH STAGE + TWO COMPARISONS WHEN $i = j$

AFTER THE FIRST STAGE 2^{k-1} TERMS

AFTER THE SECOND 2^{k-2}

NUMBER OF STAGES $k = \log n$

COMPLEXITY $2 \log n + 2 = O(\log n)$

AVERAGE CASE ANALYSIS

LINEAR SEARCH ALGORITHM.
(ASSUME x IS IN THE LIST)

n TYPES OF POSSIBLE INPUTS:

$$x = a_1 \, , \, x = a_2 \, , \, \dots \, , \, x = a_n$$

ALL ARE ASSUMED TO BE EQUALLY LIKELY

AVERAGE # OF COMPARISONS

$$\frac{3 + 5 + 7 + \cdots + (2n+1)}{n} =$$

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

$$= \frac{2(1+2+3+\cdots+n) + n}{n} = \frac{2\left[\frac{n(n+1)}{2}\right] + n}{2} =$$

$$= \frac{n(n+1) + n}{n} = n+2 \; = O(n)$$

COMPLEXITY	TERMINOLOGY
$O(1)$	CONSTANT COMPLEXITY
$O(\log n)$	LOGARITHMIC COMPLEXITY
$O(n)$	LINEAR COMPL.
$O(n \log n)$	$n \log n$ COMPL.
$O(n^b)$	POLYNOMIAL COMPL
$O(b^n)$ $(b>1)$	EXPONENTIAL COMPL
$O(n!)$	FACTORIAL COMPL

POLYNOMIAL ~ TRACTABLE (FEASIBLE)
OTHERWISE ~ INTRACTABLE

NP - SOLUTION CAN BE <u>CHECKED</u> IN POLY TIME
P - SOLUTION CAN BE <u>FOUND</u> IN POLYTIME

SATISFIABILITY OF PROPOSITIONS $\in$ NP
CHECKING $f(\bar{x}) = T$ IS LINEAR
FINDING x SUCH THAT $f(\bar{x}) = T$ IS EXPONENT.

P=NP? PROBLEM HW 2.1: 6, 2.2: 4, 10, 12