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 : INT) BEGIN
  max := a_1
  FOR i = 2 TO n
    IF max < a_i THEN max := a_i
  END
  WRITE max IS THE LARGEST ELEMENT
  END

WRITTEN IN PSEUDOCODE

INPUT/OUTPUT

DEFINEDNESS

FINITENESS

PROPERTIES: BASIC DESIRED

NUMBER OF COMPARISONS IN MAX = 2(n-1) = 2^n - 1, COMPLEXITY O(n)

THE LINEAR SEARCH ALGORITHM

PROCEDURE linear_search (x : INTEGER, a_1, a_2, ..., a_n : DISTINCT (INTS)) BEGIN
  i := 1
  WHILE i < n AND x ≠ a_i
    i := i + 1
  IF i < n THEN location := i
  ELSE location := 0
  END

GOAL: LOCATE AN ELEMENT X IN THE LIST OF DISTINCT ELEMENTS a_1, a_2, ..., a_n

NUMBER OF COMPARISONS C:

IF x IS a_i, C = 3
IF x IS NOT IN THE LIST, C = 2n - 2

COMPLEXITY IS O(n)

WORST CASE

BINARY SEARCH ALGORITHM

PROCEDURE binary_search (x : INTEGER, a_1, a_2, ..., a_n : INCREASING INTS) BEGIN
  i := 1  \{ i IS LEFT ENDPOINT OF SEARCH INTERVALS \}
  j := n  \{ j IS RIGHT ENDPOINT OF SEARCH INTERVALS \}
  WHILE i < j
    m := \lfloor (i+j)/2 \rfloor
    IF x > a_m THEN i := m + 1
    ELSE j := m
  END
  IF x = a_i THEN location := i
  ELSE location := 0
  END

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 TERMS

AFTER THE SECOND 2^{k-2} TERMS

NUMBER OF STAGES = log n

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

AVERAGE CASE ANALYSIS

LINEAR SEARCH ALGORITHM:

(ASSUME X IS IN THE LIST)

IN TYPES OF POSSIBLE INPUTS:

X = a_i, x = a_{i+1}, ..., x = a_n

ALL ARE ASSUMED TO BE EQUALLY LIKELY

AVERAGE # OF COMPARISONS

\frac{3 + 5 + 7 + ... + (2n+1)}{n}

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

= \frac{n(n+1)}{2}

= \frac{n(n+1)}{2} + n = n + 2 = O(n)
<table>
<thead>
<tr>
<th>Complexity</th>
<th>Terminology</th>
</tr>
</thead>
<tbody>
<tr>
<td>$O(1)$</td>
<td>Constant complexity</td>
</tr>
<tr>
<td>$O(\log n)$</td>
<td>Logarithmic complexity</td>
</tr>
<tr>
<td>$O(n)$</td>
<td>Linear complexity</td>
</tr>
<tr>
<td>$O(n \log n)$</td>
<td>$n \log n$ complexity</td>
</tr>
<tr>
<td>$O(n^k)$</td>
<td>Polynomial complexity</td>
</tr>
<tr>
<td>$O(2^n)$</td>
<td>Exponential complexity</td>
</tr>
<tr>
<td>$O(n!)$</td>
<td>Factorial complexity</td>
</tr>
</tbody>
</table>

**Polynomial ~ Tractable (Feasible)**  
**Otherwise ~ Intractable**

**NP** - Solution can be checked in poly-time  
**P** - Solution can be *found* in poly-time  
Satisifiability of propositions $\in$ NP  
Checking $f(x) = 1$ is linear  
Finding $x$ such that $f(x) = 1$ is exponential.

P = NP? Problem **HIV** 23, 4, 10, 12

-55-