ALGORITHMS AND COMPLEXITY

**Algorithm** — A finite sets of precise instructions for performing a computation

**Example:** Finding the maximum element

**PROCEDURE** \( \text{MAX} (a_1, a_2, \ldots, a_n : \text{integers}) \)

\[ \text{MAX} := a_1 \]

\[ \text{FOR } i = 2 \text{ TO } n \]

\[ \text{IF } \text{MAX} \leq a_i \text{ THEN } \text{MAX} := a_i \]

{MAX is the largest element}

**Written in Pseudocode**

**Input/Output**

**Definiteness**

**Finiteness**

**Properties:** Basic

**Correctness**

**Generality**

**Effectiveness**

**Desired**

Number of comparisons in \( \text{MAX} = 2(n-1) + 1 = 2n - 1 \). Complexity \( O(n) \).
**THE LINEAR SEARCH ALGORITHM**

**PROCEDURE** LINEAR SEARCH (x: INTEGER, \( q_1, q_2, \ldots, q_n \): DISTINCT INTERGERS)

\[ i := 1 \]

WHILE (\( i \leq n \) AND \( x \neq q_i \))

\[ i := i + 1 \]

IF \( i \leq n \) 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 \( q_1, q_2, \ldots, q_n \)

**NUMBER OF COMPARISONS \( C \):**

- IF \( x \) IS \( A_1 \), \( C = 3 \)
- IF \( x \) IS \( A_i \), \( C = 2i + 1 \)
- IF \( x \) IS NOT IN THE LIST \( C = 2n + 2 \)** 

**WORST CASE**

**COMPLEXITY IS \( O(n) \)**
**Binary Search Algorithm**

**Procedure** binary search (x: integer, a₁, a₂, ..., aₙ: increasing integers)

i := 1 \{ i is left endpoint of search intervals \}

j := n \{ j is right endpoint of search intervals \}

while i < j begin

m := ⌊(i+j) / 2⌋

if x > aₘ then i := m+1

else j := m

end

if x = aᵢ then location := i

else location := 0

\[ n = 2^k \quad (k = \log_2 n) \]

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 \( k = \log_2 n \)

Complexity \( 2\log_2 n + 2 = O(\log n) \)
AVERAGE CASE ANALYSIS

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

n TYPES OF POSSIBLE INPUTS:
X = a_1, X = a_2, ..., X = a_n
ALL ARE ASSUMED TO BE EQUALLY LIKELY

AVERAGE # OF COMPARISONS

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

\[
= \frac{(2\cdot1+1) + (2\cdot2+1) + (2\cdot3+1) + \cdots + (2\cdot n+1)}{n} =
\]

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

\[
= \frac{n(n+1)+n}{n} = n+2 = \Theta(n)
\]
**Complexity**

- $O(1)$
- $O(\log n)$
- $O(n)$
- $O(n \log n)$
- $O(n^k)$
- $O(b^n)$ ($b > 1$)
- $O(n!)$

**Terminology**

- **Constant Complexity**
- **Logarithmic Complexity**
- **Linear Compl.**
- **$n \log n$ Compl.**
- **Polynomial Compl.**
- **Exponential Compl.**
- **Factorial Compl.**

**Polynomial ~ Tractable (Feasible)**

**Otherwise ~ Intractable**

**NP - Solution can be checked in polytime**

**P - Solution can be found in polytime**

Satisfiability of propositions is NP.

Checking $f(x) = T$ is linear.

Finding $x$ such that $f(x) = T$ is Exponent.

**P=NP? Problem**

HW 2.1.6

HW 2.2. 4, 10, 12 - 55