Lower Bounds

Note Title 9/29/2019

§11. Lower Bounds

1] Def^n. The lower bound of a problem π is the lower bound on the complexity of all possible algorithms that solve π.

2] Def^n. efficient algorithm: is the one that has the lowest time/space complexity (usually polynomial with small power eg. $O(n \log n)$)

3] Note:

Optimal algorithm: its complexity $\equiv$ lower bound of the problem

4] Lower bounds : $\rightarrow$ trivial : by simple argument

$\rightarrow$ sophisticated : by computational model.

Examples of trivial lower bounds :

1) Find the max int in a list. $\longrightarrow \Omega(n)$

2) Multiply 2 $(n \times n)$ matrices $\longrightarrow \Omega(n^2)$

3) Find an element that is neither max nor min $\longrightarrow \Omega(1)$

5] Decision Tree Model.

e.g. Search Problem : Search for x

Alg. Linear search

Alg. Binary search (sorted list)

Decision Tree (T) for searching:

1. each node in the tree is a decision
2. Test: key $= x$?

   yes $\longrightarrow$ stop: success

   No: key $< x$ $\longrightarrow$ Go left

   key $> x$ $\longrightarrow$ Go right

   leaf-node? $\longrightarrow$ stop : fail.

test if x = key?

key

key $< x$ → Pass the control to left

key $> x$ → Pass the control to right

6] Case 1 : non-sorted list

left and right children in T do the same : Test next-key

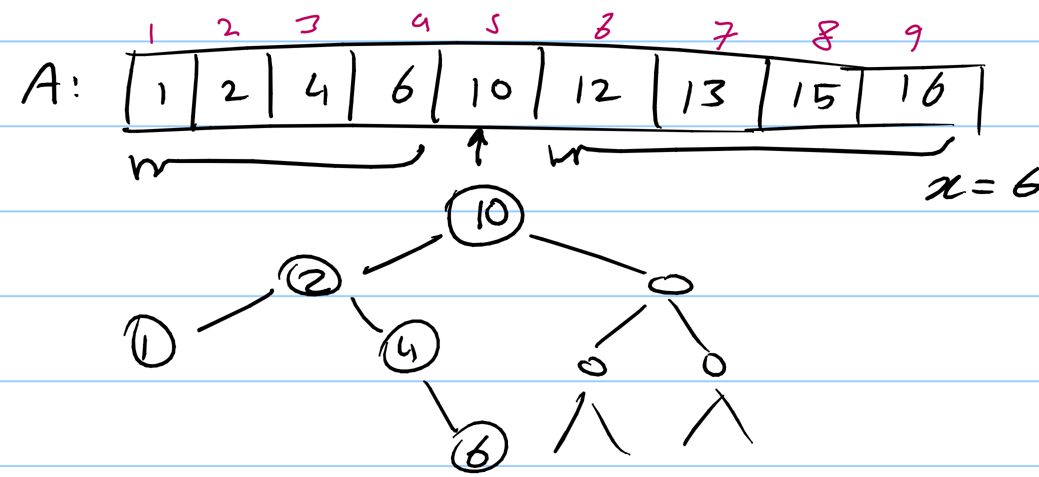
$\longrightarrow \Omega(n)$

7] $\therefore$ Algorithm: Linear Search is optimal (also efficient) for non-sorted list.

8] Case 2 : sorted - list

Left : Search the left-half

right: search the right-half



9] <u>Notes:</u> let T be the decision tree of (A, x)

1) T has m nodes $\Longrightarrow$ $m \geq n$
2) number of comparisons $=$ height $(T) + 1$
$=$ longest path from root (T) to a leaf.

3) height $(T) \geq \lfloor \log n \rfloor$

10] Thrm: Any algorithm searches a sorted list of n elements must do $\lfloor \log n \rfloor + 1$ comparisons in the worst case.

11] The lower bound on searching a sorted list is $\Omega(\log n)$

12] Alg. Binary search is Optimal.

Quiz 1:

#1.

$p = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (n+1) = (n+1)!$ $\qquad \theta((n+1)!)$

#2.

	best	Worst
Bin Search	$\Theta(1)$	$\Theta(\log n)$
insertion	$\Theta(n)$	$\Theta(n^2)$
Merge	$\Theta(n \log n)$	
Q. S.	$\Theta(n \log n)$	$\Theta(n^2)$

3.

a) $O(n)$

b) $O(1)$

c) $O(n)$

d) $O(n^2 \log n)$

$\boxed{\text{Sorted}}$ $\quad n^2$

4.

a) $f > g \quad \Omega$

$n \log n > n \quad \Omega$

$n < (\log n)! \quad O$

$n^2 \log n > n \log n \quad \Omega$

$k = \log n$

$\therefore k! = k \cdot (k-1)(k-2) \cdots)$

$\geq 2 \cdot 2 \cdot 2 \cdot 2$

$= 2^k = n$