MAT2440, Classwork35, Spring2025

ID:_____ Name:_____

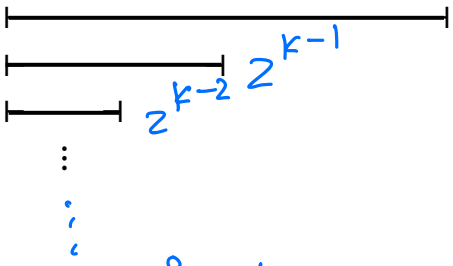
1. The time complexity analysis of the **binary search** algorithm

procedure *binary_search*(x: integer, $a_1, a_2, \cdots, a_n$: distinct integers)
$n :=$ the length of $\{a_i\}$
$i := 1$ (which is left end location)
$j := n$ (which is right end location)

while $(i \le j)$ ← 1 comparison
  m:= $\left\lfloor \frac{i+j}{2} \right\rfloor$ ← 2 operations (add, div.)
  if $x > a_m$ **then** $i := m + 1$
    else $j := m$ ← 2 operations (1 comp. 1 add)
if $x = a_i$ **then** *location* $:= i$
  else *location* $:= 0$

return *location* {either the subscript of the term that equals x, or 0 if x is not found.}

For simplicity assuming $n =$ 2^k.



	Inside the loop # of elements	# of operations
while	2^k	5
	2^{k-1}	5
	2^{k-2}	5
	⋮	⋮
	2^1	5
if (	$2^0=1$	stop
	Outside the loop	1

} K times

Therefore, the time complexity is

$n = 2^k \Rightarrow k = \log_2(n)$

$f(n) =$ $5k+1$ $= 5\cdot\log_2(n)+1 \sim 5\cdot\log_2(n) = 5\cdot\frac{\log(n)}{\log(2)} = O(\log(n))$

change base property

which is logarithmic complexity.

2. Binary search ($O(\log\ n)$) is more efficient than linear search ($O(n)$).

3. The worse-case time complexity analysis of the **sorting** algorithm: **Bubble Sort**.

procedure *bubblesort*($a_1, a_2, \cdots, a_n$: real numbers with $n \geq 2$)
$n :=$ the length of $\{a_i\}$
for $i := 1$ **to** $n - 1$
  for $j := 1$ **to** $n - i$
    if $a_j > a_{j+1}$ **then** interchange a_j and a_{j+1}
$\{a_1, a_2, \cdots, a_n$ is in increasing order$\}$

$i = 1$, $j = 1$ to _____ ←_____ operations
$i = 2$, $j = 1$ to _____ ←_____ operations
⋮ ⋮ ⋮
$i =$ ___, $j = 1$ to ___ ←_____ operations

Total operations:

The time complexity is $f(n) =$ _____ $= O($____$)$ which is _____ complexity.

4. The worse-case time complexity analysis of the **sorting** algorithm: **Insertion Sort**.

procedure *insertionsort*($a_1, a_2, \cdots, a_n$: real numbers with $n \geq 2$)
$n :=$ the length of $\{a_i\}$
for $i := 2$ **to** n
  $j := 1$
  while ($a_i > a_j$ and $i > j$)
    $j := j + 1$
  $m := a_i$
  for $k := 0$ **to** $i - j - 1$
    $a_{i-k} := a_{i-k-1}$
  $a_j := m$
$\{a_1, a_2, \cdots, a_n$ is in increasing order$\}$

$i = 2$,	$a_2 > a_1, 2 > 1, j =$ _____	←_____ operations
$i = 3$,		←_____ operations
$i = 4$,		←_____ operations
⋮		
$i = n$,		←_____ operations

Total operations:

The time complexity is $f(n) =$ $= O($____$)$ which is _____ complexity.