FRESNO STATE

Discovery. Diversity. Distinction.

CSCI 115 – Algorithms and Data Structures

Midterm Exam 1

Dr. M. Pirouz

Fill in the Blank

Time complexity of MAX-HEAP-INSERT function is O(lgn)

In array representation of a heap, the formula to calculate Parent of A[i] is A[$\lfloor i/2 \rfloor$]

In a heap, the range of leaves is A[($\lfloor n/2 \rfloor$+1) .. n]

Quick sort best case time complexity is nlogn and worst case is O(n^2)

The max number of internal nodes in a complete binary tree with 57 nodes is 28

Find a closed-form solution to the sum $\sum_{i=0}^{n} 2i - 2$ as a polynomial in n. Show your work and write the coefficients of your answer in the blanks at the bottom.

$$\sum_{i=1}^{n} 2i - \sum_{i=1}^{n} 2$$

$$= \frac{2n(n+1)}{2} - 2n = n^2 + n - 2n = n^2 - n$$

Considering that the sum in the question started from 0 instead of 1 the calculation will change to the following:

$$\sum_{i=0}^{n} 2i - \sum_{i=0}^{n} 2$$

$$= \frac{2n(n+1)}{2} - 2(n+1) = n^2 - n - 2$$

Find a closed-form solution to the sum $\sum_{i=0}^{n} 6i - 3$ as a polynomial in n. Show your work and write the coefficients of your answer in the blanks at the bottom.

$\sum_{i=1}^{n} 6i - \sum_{i=1}^{n} 3$

$= \frac{6n(n+1)}{2} - 3n = 3n^2 + 3n - 3n = 3n^2$

Considering that the sum in the question started from 0 instead of 1 the calculation will change to the following:

$\sum_{i=0}^{n} 6i - \sum_{i=0}^{n} 3$

$= \frac{6n(n+1)}{2} - 3(n+1) = 3n^2 - 3$

Determine the Big-O notation for the running time of the following codes:

```
int i = 0, j = 0;
for(; i < n; ++i){
    while(j < n && arr[i] < arr[j]){
        j++;
    }
}
```

$O(n^2)$

```
for(i=0, i <n , i++){
    for(j=1, j <log(i) , j++){
        c=i+c;
    }
}
```

O(nlogn)

For the merge sort:

1. [5 pts] Complete the algorithm from the exhibit here.
2. [6 pts] Provide the <u>recursive equation</u>
3. [4 pts] Provide the time complexity

Alg.: MERGE-SORT(A, p, r)

if $\boxed{\text{a}}$

then $q \leftarrow \lfloor \boxed{\text{b}} \rfloor$

MERGE-SORT(A, p, q)

MERGE-SORT($\boxed{\text{e}}$)

MERGE(A, p, q, r)

Method/Action:
Check for base case
$\boxed{\text{c}}$
$\boxed{\text{d}}$
Conquer
Combine

- Initial call: MERGE-SORT(A, 1, n)

a = p<r

b = p+r/2

c=divide

d=conquer

e=(A,q+1,r)

2) 0(1)+2T(n/2)+O(n)

3)nlogn

Solve the following equation using the iteration method:

$T(n) = 2T(n/2) + n$

Assume: $n = 2^k$

$T(n/2) = n/2 + 2T(n/4)$

$$
\begin{aligned}
T(n) &= n + 2T(n/2) \\
&= n + 2(n/2 + 2T(n/4)) \\
&= n + n + 4T(n/4) \\
&= n + n + 4(n/4 + 2T(n/8)) \\
&= n + n + n + 8T(n/8) \\
\ldots\ &= in + 2^iT(n/2^i) \\
&= kn + 2^kT(1) \\
&= n\lg n + nT(1) = \Theta(n\lg n)
\end{aligned}
$$

Find the asymptotic upper bound of the following recurrence using master method: **$T(n) = 3T(n/4) + n\lg n$**

$a = 3$, $b = 4$, $\log_4 3 = 0.793$

Compare $n^{0.793}$ with $f(n) = n\lg n$

$f(n) = \Omega(n^{\log_4 3+\varepsilon})$ Case 3

Check regularity condition:

$$3*(n/4)\lg(n/4) \le (3/4)n\lg n = c*f(n),\ c=3/4$$

$\Rightarrow T(n) = \Theta(n\lg n)$

Case 1: if $f(n) = O(n^{\log_b a-\varepsilon})$ then $T(n) = \Theta(n^{\log_b a})$

Case 2: if $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a}\lg n)$

Case 3: if $f(n) = \Omega(n^{\log_b a+\varepsilon})$ then $T(n) = \Theta(f(n))$!

$af(n/b) \le cf(n)$ for some $c < 1$

Find the asymptotic upper bound of the following recurrence using master method: **T(n) = 4T(n/2) + n3**

a = 4, b = 2 => nlogba = n^2; f (n) = n^3.

CASE 3: f (n) = = $\Omega(n^{\log_2 4+\varepsilon})$ for $\varepsilon = 1$

and 4(n/2)^3 $\leq$ cn^3 (reg. cond.) for c = 1/2.

$\therefore$ T(n) = Θ(n^3).

Case 1: if $f(n) = O(n^{\log_b a - \varepsilon})$ then $T(n) = \Theta(n^{\log_b a})$

Case 2: if $f(n) = \Theta(n^{\log_b a})$, then $T(n) = \Theta(n^{\log_b a} \lg n)$

Case 3: if $f(n) = \Omega(n^{\log_b a + \varepsilon})$ then $T(n) = \Theta(f(n))$!

$af(n/b) \leq cf(n)$ for some c < 1

For the following expression, find C and n_0 values such that
$530\ n^2+75n=O(n^2)$, $550\ n^2+75n=O(n^2)$, $532\ n^2+75n=O(n^2)$

$530\ n^2+75n <= 530n^2 +75n^2 <= Cn^2$

$605n^2 <= Cn^2$ ➔ C=605 , n0=1

Compare complexities for f(n) and g(n) using either >, <, or =. Include your justification and show your thought process.

	f(n)	$\prec$ or $\succ$ or $\sim$	g(n)
a	n^n	$n \cdot n \cdot n, \ldots \succ n \cdot (n-1) \cdot (n-2) \ldots$ $>$	n!
b	n^2	$\sim$ $4^{(log^n)} = n^{(log^4)} = n^2$	$4^{(log\, n)}$
c	n log n	nlogn > n , 1 > 10/11 >	$n^{10/11}$
d	log 10	Constants so $\sim$	10

Heap-sort pseudo-code is given below. What would happen if we remove line 4 from the pseudocode?

HEAPSORT(A)
1 BUILD-MAX-HEAP(A)
2 **for** *i = A.length* **downto** 2
3 exchange A[1] with A[i]
4 *A.heapSize = A.heapSize – 1*
5 MAX-HEAPIFY(A, 1)

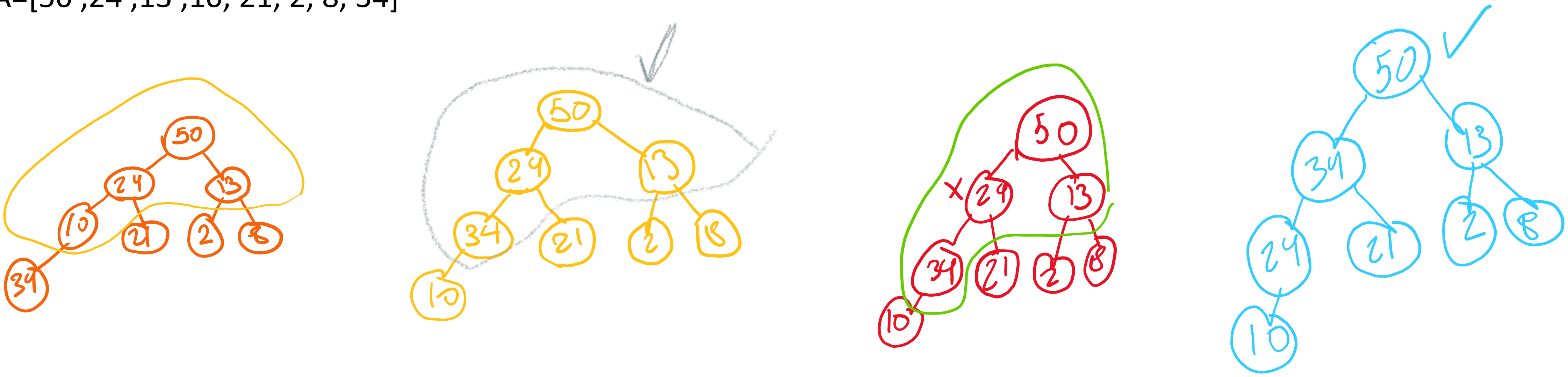
Removing line 4 from this heap-sort pseudo code causes the Max-heapify function to include the recently sorted elements at the end of the array in the range of available nodes to compare against, which will sabotage all the sorting done from exchanging the maximum at root with the last position of the heap

For the following C++ code find and write the recurrence relation. You need to model the runtime of function "Func" in terms of n. (only the recurrence relation in terms of n, No output of the code or final runtime analysis is required)

```
void Func(int L[], int s, int e){
    if (s < e) {
        i=s-1;
        for (int j = s; j <= e - 1; j++) {
            if (A[j] <= x){
                i++;
                swap (&L[i], &L[j]);}}
          swap (&L[i + 1], &L[e]);
     int k = i+1;
     Func(L, s, k - 1);
     Func(L, k + 1, e);
}}
```

Use the BUILD-MAX-HEAP algorithm to Build a Max Heap for the following array A (show steps of your work):

A=[50 ,24 ,13 ,10, 21, 2, 8, 34]



function $f(n)=O(n)$ and function $g(n)=O(n^3)$ and $h(n)= n\log n$. Indicate if following statements are True (T), False (F) or semi true (ST).

$f(n)= O(g(n))$ ST

$h(n)=\Omega(f(n))$ T

$f(n) + g(n)= O(g(n))$ ST

$f(n) .g(n)= O(h(n))$ F

Questions!