O
G
D
O
O
O
O
D

Instruction: Please answer all questions. Total score is 78 points.

1. *[3 pts]* Show that $3n^2 - 6n + 8 = \Theta(n^2)$

$3n^2 - 6n + 8 \geq n^2$ — (*) | $3n^2 - 6n + 8 \leq n^2$ — (**)

$3n^2 \geq n^2 \quad \forall n$

$-6n \geq 0 \quad \forall n$

$8 \geq 0 \quad \forall n$

$3n^2 - 6n + 8 \geq n^2$

$3n^2 \leq 3n^2 \quad \forall n$

$-6n \leq n^2 \quad \forall n$

$8 \leq n^2 \quad \forall n \geq 3$

$3n^2 - 6n + 8 \leq 5n^2$

$\therefore \; 3n^2 - 6n + 8 = \Theta(n^2) \quad \forall n \geq 3$

2. *[3 pts]* Show that $n \log n = O(n^2)$

$n \leq n \quad \forall n$

$\log n \leq n \quad \forall n$

$n \log n \leq n^2 \quad \forall n$

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

3. *[3 pts]* Suppose $f(n) = O(g(n))$ and $\forall n \; f(n) > 0$. Show that whether or not $\sqrt{f(n)} = O(\sqrt{g(n)})$ and explain your answer.

4. *[3 pts]* Show the running time of method **F1**. Find the smallest big-O notation as a function of *n*. Note that abs(i) return absolute value of i.

```
int F1 (int n)
{   int sum=0, i, j
    for i = -n to n
        j = 0;
        while (j < abs(i))
            sum = sum+1;
            j = j+2;
        end
    end
    return sum;
}
```

$2n+1$

n

5. *[3 pts]* Show the running time of method **F2**. Find the smallest big-O notation as a function of *n*.

```
int F2 (int n)
{   int sum=0, i, j
    for i = 1 to n-1
        j = i;
        while (j < n)
            sum = sum+1;
            j = j+i;
        end
    end
    return sum;
}
```

for i = 1 to n-1 $\rightarrow O(n)$

while (j < n) $\rightarrow O(n)$

i	j	
1	$1, 2, 3, 4, \ldots, n-1$ $\Rightarrow$	$n-1$
2	$2, 3, 4, 5, \ldots, n-1$	$n-2$
		$n-3$
$\vdots$	$\vdots$	$\vdots$
$n-1$	$n-1$	1

Runtime $= \frac{(n-1)n}{2} = \frac{n^2-n}{2} = O(n^2)$ #

6. *[3 pts]* Show the running time of method **F3**. Find the smallest big-O notation as a function of *n*.

```
int F3 (int n)
{   int sum=0, i=0, j=0
    while (i < n)
        while (j < i)
            sum = sum+1;
            j = j+1;
        end
        i = i+10
    end
    return sum;
}
```

$f(n) \geq g(n)$

7. *[3 pts]* Suppose $f(n) = \Omega(g(n))$ and $\forall n \; f(n) > 0$. Show that whether or not $\log(f(n)) = O(\log(g(n)))$ and explain your answer.

Let $g(n) = 1$, $f(n) \geq g(n)$

$\log(g(n)) = 0 < \log(f(n))$

$\therefore \log(f(n)) \neq O(\log(g(n)))$

8. *[4 pts]* Suppose $T(n) = 4T(n-2) + 2^n$ and $T(1) = 1$. Find the smallest big-O of $T(n)$ and show your work. For convenience, you may assume that n is an odd integer

$T(n) = 4\,T(n-2) + 2^n$ $k=1$

$4T(n-4) + 2^{n-2} = 4^2\,T(n-4) + 2^n\left(1 + \frac{1}{2^2}\right)$ $k=2$

$4^2\left(4T(n-6) + 2^{n-4}\right) + 2^n\left(1 + \frac{1}{2^2}\right) = 4^3\,T(n-6) + 2^n\left(1 + \frac{1}{2^2} + \frac{1}{2^4}\right)$ $k=3$

$= 4^k T(n-2k) + 2^n\left(1 + \frac{1}{2^{2(1)}} + \frac{1}{2^{2(2)}} + \dots + \frac{1}{2^{2(k-1)}}\right)$

$T(n) = 4^k T(n-2k) + 2^n\left(\frac{1-\left(\frac{1}{4}\right)^k}{1-\frac{1}{4}}\right)$

$k = \frac{n+1}{2}$

$= 4^{\frac{n+1}{2}}\,\cancel{T(1)}^{1} + 2^n\left(\frac{1-\left(\frac{1}{4}\right)^{\frac{n+1}{2}}}{\frac{3}{4}}\right)$

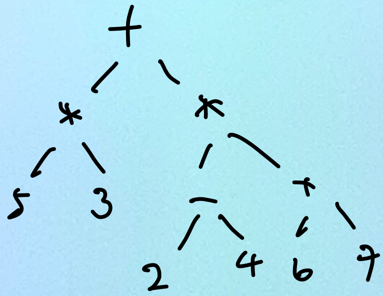
$= 2^{n+1} + 2^n\left(\frac{1-\frac{1}{2^{n+1}}}{\frac{3}{4}}\right)$ $\qquad \frac{2^n - \cancel{\frac{1}{2}}}{\cancel{\frac{3}{4}}}$

$= O(2^n) + O(2^n) = O(2^n)$ ##

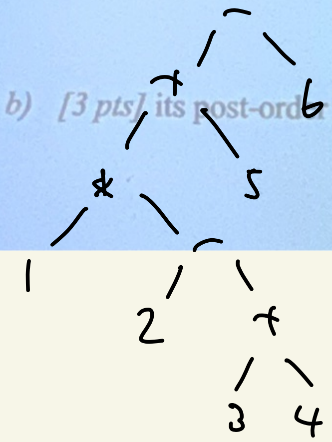
9. *[4 pts]* Suppose $T(n) = 2T\left(\frac{n}{4}\right) + \sqrt{n}$ and $T(1) = 1$. Find the smallest big-O of $T(n)$ and show your work. For convenience, you may assume that $\exists k \in \mathbb{Z},\ n = 4^k$.

10. *[9 pts]* Suppose an *Expression Tree* is a tree that all internal nodes are operations and all leave nodes are number. Find the expression tree when ….

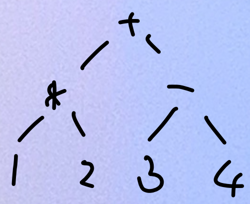
a) *[3 pts]* its pre-order is + * 5 3 * – 2 4 + 6 7



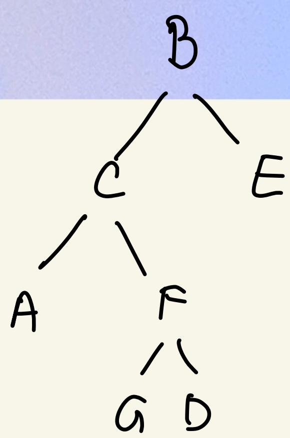
b) *[3 pts]* its post-order is 1 2 3 4 + – * 5 + 6 –



c) *[3 pts]* Answer two different expression tree if its in-order is 1 * 2 + 3 – 4



11. *[6 pts]* Find a tree when is pre-order is B C A F G D E and its post-order is A G D F C E B.



12. *[6 pts] Suppose that you already have implemented Stack*, and now you want to use *Queue*.
Is it possible to use 2 stacks to imitate a queue? Explain your answer.

13. *[4 pts] Suppose that you already have implemented Queue,* and now you want to use *Stack*.
Is it possible to use 2 queues to imitate a stack? Explain your answer.

14. *[6 pts]* Suppose you have a linked list named *head*. You want to find the first non-*head* node that its value is smaller than the value of the previous node. Please define the linked-list and write your function to find and remove that node (if any).

15. *[8 pts]* Suppose you have a linked list named *head*. Unfortunately, you are not sure whether there is a loop inside this linked list. Give an idea (or write a program) to find whether linked-list has a loop. When …

Note: if you answer *b)*, you do not need to answer *a)*

a) *[4 pts]* the linked-list has at most 100 nodes.

16. *[10 pts]* Let array $A = \{a_1, a_2, a_3, \dots, a_n\}$. For any index i, find the largest index $k < i$ that $a_k > a_i$. Formally, find the index k such that $a_k > a_i$ and $\forall j \in \{k+1, k+2, \dots, i\}, a_j \le a_i$.