1) a) For the code segment shown below.

(a) (4 pts) Find the Big-Oh order of this code segment in terms of \( n \). Provide justification.

\[
\sum_{i=1}^{2n} \sum_{j=1}^{n} 1 = \sum_{i=1}^{2n} n = n \cdot 2n = O(n^2)
\]

(b) (6 pts) Determine a summation representing the final value of \( x \). Solve for the closed form solution in terms of \( n \).

```c
int x = 0;
for (i = 1; i <= (2*n); i++) {
    for (j = 1; j <= n; j++) {
        if (j < i)
            x = x + 1;
    }
}
```

\[
\sum_{i=1}^{2n} \frac{\sum_{j=1}^{n} 1 - \sum_{j=1}^{n} j}{2} = \sum_{i=1}^{2n} \frac{n - \frac{n(n+1)}{2}}{2}
\]

\[
2n^2 - \frac{n^2 + n}{2} = \frac{3n^2 - n}{2}
\]

or \( x \) gets incremented \( 1, 2, 3 \ldots \frac{(n-1)n}{2} \)

then \( x \) gets incremented \( n \) times

the last \( n \) iterations of the outer loop

\[
O(n^2 + n^2) = \frac{3n^2 - n}{2}
\]

c) An algorithm runs in \( O(\sqrt{n}) \) time. When the algorithm is run with an input size of 14900, it takes 7 seconds to complete. How long will it take to complete on an input size of 59600?

\[
C\sqrt{n} = \text{time} \quad C\sqrt{14900} = 7\text{ms} \quad C = \frac{7}{\sqrt{14900}}
\]

\[
\frac{7}{\sqrt{59600}} = 7\cdot \frac{\sqrt{14900}}{\sqrt{14900}} = 14\text{ms}
\]

d) Let \( T(n) \) represent the best case run-time of a Quick Sort of \( n \) elements. Write down a recurrence relation that \( T(n) \) satisfies that is based on the standard recursive implementation of the sort.

best case we evenly divide partition the list in half

\[
T(n) = 2T\left(\frac{n}{2}\right) + O(n)
\]

2 recursive calls time it took to partition to Quick sort on half the elements
This question was not specific enough, make the following change:

6) Write a recursive function that pushes all even values into a Stack and enqueues all odd values into a Queue. Assume S and Q are initialized.

Starting with the least values first, i.e. an In Order traversal.

```
 struct tree_node {
   int data;
   struct tree_node* left;
   struct tree_node* right;
 };

 void int question6(struct tree_node *root, Stack *S, Queue *Q) {
   if (root == NULL)
     return;

   question 6(root->left, S, Q);
   if (root->data % 2 == 0)
     push(S, root->data);
   else
     enqueue(Q, root->data);

   question 6(root->right, S, Q);
 }
```

After the function call question6(root, S, Q), where root points to tree_node 30, and S and Q are originally empty. What are the contents of S and Q?

S: 6 8 22 28 30 32 36 42
Q: 3 15 23 25 41 43
a) Give the adjacency matrix of the graph above:

\[
\begin{array}{cccccc}
  & U & C & F & R & O & X \\
U & 0 & 1 & 0 & 0 & 0 & 0 \\
C & 1 & 1 & 0 & 0 & 1 & 0 \\
F & 0 & 1 & 0 & 0 & 0 & 0 \\
R & 0 & 0 & 0 & 0 & 1 & 0 \\
O & 0 & 1 & 0 & 1 & 0 & 0 \\
X & 0 & 1 & 0 & 0 & 0 & 0 \\
\end{array}
\]

b) Give the Depth First Search traversal of the graph above, starting with U and if there is a choice between vertices choose in alphabetical order.

DFS: \[U \rightarrow C \rightarrow F \rightarrow R \rightarrow O \rightarrow X\]

c) Give the Breadth First Search traversal of the graph above, starting with U and if there is a choice between vertices choose in alphabetical order.

BFS: \[U \rightarrow C \rightarrow F \rightarrow R \rightarrow X \rightarrow O\]