Q.1. Determine recurrence equation and solve it with Master Theorem and recursion tree method of the following algorithm.

```
Algo (A, start, end) &
    if (end - start + 1 ≤ 4) return 0
    Sum = 0
    for i = start to end
        for j = i to end
            Sum = Sum + A[j]
    m = \left\lceil \frac{start + end}{4} \right\rceil
    return Algo (A, start, m) + Algo (A, m+1, 2m) +
         Algo (A, 2m+1, 3m) + Algo (A, 3m+1, end)
```

\textbf{Solution}:
\[
T(n) = \sum_{i=0}^{4} 4T(n/4) + \Theta(n^2)
\]
\[
\Theta(1)
\]

\textbf{Master Theorem}:
\[a = 4, \quad b = 4, \quad f(n) = n^2\]
\[c < 1, \quad n^{\log_b(a)} = n^{\log_{4}(4)} = n^1\]
\[a f\left(\frac{n}{b}\right) \leq c f(n) \quad \frac{f(n)}{n^{\log_b(a) + \varepsilon}} = O\left(n^{\log_{4}(4) + \varepsilon}\right) \quad \varepsilon = 1\]
A student needs to select the classes for the semester. He has a set of $C = \{c_1, \ldots, c_n\}$ possible classes. Class $c_i$ requires effort $e_i$ and has $q_i$ credits. He has to select classes for at least $Q$ credits. He wants to select the set of classes that requires minimum efforts.

1) Propose two greedy approaches and show with counter examples that they do not work.

2) Prove that if $q_i = 1 \forall i$ the greedy algorithm that select at each iteration the class with least effort is optimal.

Solution: \[
\text{Min-effort} (C, Q) \{
    T = \emptyset \quad \text{// sum of credits of selected classes}
    S = \emptyset \quad \text{// sum of efforts of selected classes}

    \text{while} \ (T < Q) \{
        C_k = \arg \min_{C_i \in C \setminus S} e_i
        S = S \cup \{e_k\}
        T = T \cup \{q_k\}
    \}
\]
We can use the min-effort algorithm:

- **Termination**: The algorithm terminates as soon as $|\Phi|$ classes are selected.

- **Proposed Solution**: Is included in the optimal solution. Suppose $S_h$ is the partial solution at iteration $h$.

**Base case**: $h = 0$, i.e., $S_0 = \emptyset = S_0 \subseteq S^*$

Empty set is subset of any set.

**Inductive hypothesis**: $S_h \subseteq S^*$, we need to prove $S_{h+1} \subseteq S^*$.

**Inductive step**: Suppose class $c_k$ is selected at $h+1$ iteration.

If ($c_k \in S^*$) then $S_{h+1} = S_h \cup \{c_k\} \subseteq S^*$

and we are done!!
if \([C_x \notin S^*]\), then
we know \(|S_{n+1}| < |S^*|\).
So there exist a class \(C_j\) in \(S^*_{S_{n+1}}\), for

which effort is minimum, i.e.

\[
C_j = \arg \min_{e_i} e_i \quad \forall e_i \in S^* \setminus S_{n+1}
\]
\[ S^\# = (S^* \setminus \{ e_j \}) \cup \{ c_k \} \]

1) \( |S^\#| = |S^*| \)

2) The algorithm could have picked \( c_j \) since it was not in \( S_h \), but picked \( c_k \).

\[ \Rightarrow e_k \leq e_s \]

\[ \Rightarrow \sum_{c_i \in S^*} e_i \leq \sum_{c_i \in S^*} e_i \]

\[ \Rightarrow S^\# \text{ is optimal too.} \]

The solution is optimal.

8 iterations \( \Rightarrow S_8 \) is the final solution.

we know that \( S_8 \leq S^* \) optimal.

\[ \Rightarrow |S_8| \leq |S^*| \]

\[ \Rightarrow \sum_{c_i \in S_8} e_i \leq \sum_{c_i \in S^*} e_i \]

\[ \Rightarrow S_8 \text{ is optimal.} \]