1. Let \( f(n) \) and \( g(n) \) be asymptotically positive functions. Prove or disprove the conjecture: [20 points]

\[
f(n) = O(g(n)) \text{ implies } 2^{f(n)} = O(2^{g(n)})
\]

(Hint: For a given function \( g(n) \), we denote by \( O(g(n)) \) the set of functions

\[
O(g(n)) = \{ f(n) : \text{ there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \leq f(n) \leq cg(n) \text{ for all } n \geq n_0 \}.
\]

We cannot get \( f(n) = O(g(n)) \Rightarrow 2^{f(n)} = O(2^{g(n)}) \)
because \( 2n = O(n) \), but \( 2^n \neq O(2^n) \)

2. Use the master method to give tight asymptotic bound for the following recurrence, and verify the bound by the substitution method. [20 points]

\[
T(n) = 2T(n/4) + n.
\]

Based on master method, \( a = 2, b = 4, f(n) = n \).

There exists a \( \varepsilon = 1/2 \) to satisfy case 3 so that

\[
f(n) = O(n^{\log_4 2 + \varepsilon}) = O(n^{1/2 + \varepsilon}) = O(n)
\]

Guess \( O(n) \)

Assume that \( T(k) \leq ck \) for \( k < n \).

\[
T(n) = 2T(n/4) + n \
\leq c(n/4) + n = (1 + c/2) n \leq cn,
\]

for \( c > 2 \) and \( n \geq 1 \)

3. The following procedure implements quicksort, and answer questions:

```
PARTITION(A, p, r)
1  x = A[r]
2  i = p - 1
3  for j = p to r - 1
4      if A[j] \leq x
5          i = i + 1
6      exchange A[i] with A[r]
7  exchange A[i + 1] with A[r]
8  return i + 1

QUICKSORT(A, p, r)
1  if p < r
2      q = PARTITION(A, p, r)
3      QUICKSORT(A, p, q - 1)
4      QUICKSORT(A, q + 1, r)
```

(1) Illustrate the operation of PARTITION on the array \( A = \{13; 19; 9; 5; 14; 7; 4; 21; 6; 14\} \) by redrawing the array right after there is a swap of a pair of elements. [15 points]
Suppose that the partitioning algorithm always produces a 2-to-1 proportional split, what is the recurrence (function)? [15 points]

\[ T(n) = T(n/3) + T(2n/3) + cn \]

Use recursion tree to obtain the recurrence’s upper bound. (Hint: it should be the same as the best worst-case running time for comparison sorting). [15 points]

\[ T(n) = cn \log_2 n = O(n \log n) \]

Use the substitution method to prove the upper bound. [15 points]

Guess \( O(n \log n) \)

Assume that \( T(k) \leq a \cdot k \log k \) for \( k < n \).

\[
T(n) = T(n/3) + T(2n/3) + cn \\
\leq a \cdot (n/3) \cdot \log(n/3) + a \cdot (2n/3) \cdot \log(2n/3) + cn \\
= an \log n + (c + a/3 \cdot \log(4/27))n \\
\leq an \log n \\
\text{For } n > 1 \text{ and } c + (a/3) \cdot \log(4/27) < 0
\]