CS 692  Capstone Exam  Algorithms  Fall 2020: Choose any 2 of the 3 problems.

1) A “full node” in a binary tree has either no children or exactly 2 children. Given a binary tree, write a function that returns true if all nodes are full nodes. Otherwise return false.
Notes: The function should have just one argument, a pointer to the root.
   No global variables may be used.
   No additional functions may be defined.

2) For each function with input argument $n$, determine the asymptotic number of “fundamental operations” that will be executed. Note that $f_c$ is recursive. Choose each answer from among the following. You do not need to explain your choices.
\[ \Theta(1) \quad \Theta(\log n) \quad \Theta(n) \quad \Theta(n \log n) \quad \Theta(n^2) \quad \Theta(n^2 \log n) \quad \Theta(n^3) \quad \Theta(2^n) \quad \Theta(n!) \]

a) void fa(int n) {
   for(i = 1; i < n; i++) {
      for(j = i; j <= n; j++)
         Perform 1 fundamental operation;
   }
   for(k = 1; k <= n; k++)
      Perform 1 fundamental operation;
}

b) void fb(int n) {
   for(i = 1; i <= n; i = 3*i)
      Perform 1 fundamental operation;
}

c) void fc(int n) {
   if (n > 1) {
      fc(n/2);
      fc(n/2);
      Perform 1 fundamental operation;
   }
}

d) For part d), a fundamental operation will be a comparison of array element types.

void hybrid_sort(element a[], int n){
   // Put a[0]..a[n-1] into ascending order.
   if (n < 100)
      Perform selection sort;
   else
      Perform mergesort;
}

3) Solve the recurrence relation $T(n) = T(n/2) + 3n$ where $T(1) = 0$ and $n = 2^k$
for a nonnegative integer $k$. Your answer should be a precise function of $n$ in closed form (i.e.,
resolve all sigmas and ...’s). An asymptotic answer is not acceptable. Justify your solution.