Answer **ANY TWO** of the following three questions:

1. (DS #1) Consider two linked lists of integers, S1 and S2, both sorted in ascending order. Write a function called *union* that, given S1 and S2, returns a new list S3 that is the union of the two lists in ascending order.

   Example:

   S1 $\rightarrow 2 \rightarrow 4 \rightarrow 5 \rightarrow 8 \rightarrow 11$

   S2 $\rightarrow 3 \rightarrow 5 \rightarrow 9 \rightarrow 11$

   S3 $\rightarrow 2 \rightarrow 3 \rightarrow 4 \rightarrow 5 \rightarrow 8 \rightarrow 9 \rightarrow 11$

2. (DS #2) An *interior node* of a tree has at least one child (and is therefore not a leaf). Given a binary tree, write a recursive function that returns the number of its interior nodes. Code in the language of your choice and include declarations for your data structures. Do not code any other functions.

3. (Analysis) Solve the recurrence relation:

   $$T(n) = T(n/2) + n \text{ for } n > 1; T(1) = 1$$

Your answer should be a precise function of n in closed form.