1. **StupidSort:** Consider the following algorithm, called stupidsort, which takes two indices a and b, and sorts the elements of an array between positions a and b.

```
StupidSort(a, b)
   <Some base case>
   n = ceiling(2(b+1-a)/3)
   StupidSort(a, a+n)
   StupidSort(b-n,b)
   StupidSort(a, a+n)
```

   That is, it recursively sorts the first $\frac{2}{3}$ of the input, then the last $\frac{2}{3}$, and then the first $\frac{2}{3}$ again. Prove that StupidSort correctly sorts its input. Also calculate the running time of the StupidSort algorithm. What would happen if we replaced the ceiling with floor on the first line of the input?

2. **Cheating Students:** There are n naughty students, and each naughty student $s, 1 \leq s \leq n$, has an answer to a distinct upcoming exam problem p_s. They would like to share their answers before the exam, but only two students can share information at a time since a meeting of more than two would arouse suspicion. Assume that if two students meet to exchange solutions that they each know, then after the meeting, they each know the union of the solutions known by the two. Since the exam is quickly approaching, they would like the quickest method for sharing, as measured by the minimum number of pairwise meetings total before all students know all solutions. Find an algorithm to do this until they all know all the solutions, your algorithm should use as small number of pairwise meetings as possible.

3. **Rotating Pictures:** An assignment for a Computer Graphics course requires students to write a program that rotates pictures by 90 degrees. Assume that the picture is square, and has size $2^n \times 2^n$ pixels. Give a divide-and-conquer algorithm that makes 4 recursive calls, and 5 *block transfers*. A block transfer is a library routine that copies a square block of pixels from one location to another; this can be implemented fairly quickly.