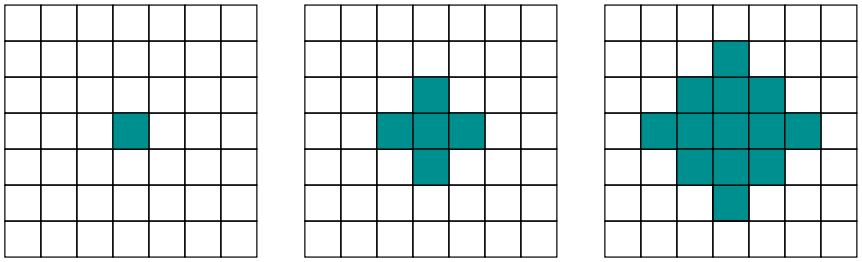
INSTRUCTIONS:

Write or type your answers on paper. Hand in your answers in class on the due date shown above. Attempt all problems.

Problem 1: **[4 pts]** (Exercise 11, Section 2.3) How many one-by-one squares are generated by the algorithm that starts with a single square and, on each of its n iterations, adds new squares all around the outside? The configurations for iterations 1, 2, and 3 are illustrated below.



Problem 2: **[5 pts]** (Exercise 3, Section 3.2) A firm wants to determine the highest floor of a building with n floors from which a gadget can fall with no impact on its functionality. The firm has two identical gadgets for this purpose. Describe in pseudocode an algorithm that requires, in the worst case, $O(\sqrt{n})$ gadget drops to solve the problem and argue its complexity.

Problem 3: **[4 pts]** Consider the brute force string matching algorithm discussed in Section 3.2. If n is the size of the text and m the size of the pattern, describe a class of problem instances for which the algorithm makes at least $\Omega(nm)$ character comparisons. Explain your answer.

Problem 4: **[7 pts]** (Exercise 9, Section 4.2) Let $A[0..n-1]$ be an array of n distinct real numbers. A pair $(A[i], A[j])$ is called an inversion if $i < j$ and $A[i] > A[j]$. Describe an algorithm with $O(n \log n)$ worst case complexity, based on divide and conquer, to determine the number of inversions in the array. You are allowed to change the order of the elements in the array. Give an analysis of the algorithm.