This homework covers developing divide-and-conquer algorithms. It is due in class Friday, April 4.

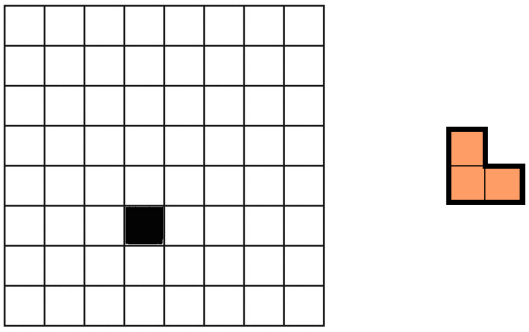
Write your solutions carefully — your work should be neat, readable, organized, and polished. As with writing a paper, you will likely need at least two drafts — the first draft can be rough (similar to what was written in class) as the focus is on going through the steps and finding a solution, while in the final draft the focus is on clarity of explanation and the quality of writing (similar to what was written after class). While it is not required, you are encouraged to type your work so that the revisions for the final draft are easier to make.

See the Policies page on the course website for information about revise-and-resubmit, late work, and academic integrity as it applies to homework.

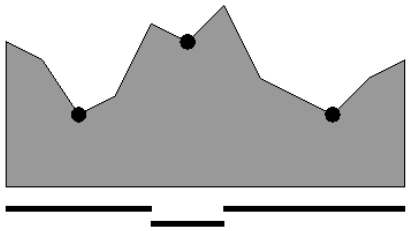
For each of the following problems, develop a divide-and-conquer algorithm to solve the problem using the process discussed in class. Give each of the steps in the template — don't just give an algorithm. This is not a research task to look up the solution — the point here is to understand and be able to apply the process discussed in class to develop an algorithm yourself.

1. Given a collection of L-shaped tiles and an $n \times n$ board with one square blacked out, find an arrangement of tiles that covers all but the blacked-out square. (Each L-shaped tile covers three board squares.) Tiles may not overlap each other. You can assume n is a power of 2.

   An example of a board with one blacked-out square is shown below, along with an L-shaped tile.



2. A *watershed* is the area of land that drains into a particular water system — the Seneca Lake watershed, for example, is the area from which water drains into Seneca Lake. The picture below shows a 2D version of this concept — the landscape is shown in gray, the points where water falling on the landscape would collect are shown with black dots, and the watershed for each of the collection points is shown by the span of the black lines below the landscape. Given an array of n elevation values, report the collection points (as the array index of that point) and their watersheds (as a span of array indexes).



3. In an array of numbers, an inversion is a pair of numbers which are out of order (according to increasing value). For example, an array containing `1 2 5 4 3` would have three inversions (5,4), (5,3), (4,3). Count the number of inversions in an array of length n.