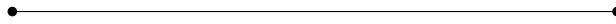


Worksheet: A Random Coastline

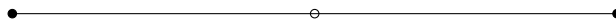
This is a programming exercise, but you and your group should start writing a program on paper. At the end it will become a running MATLAB/OCTAVE code.

N-level Coastline Algorithm.

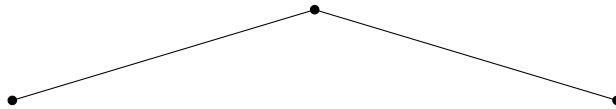
- (1) Start with one horizontal line segment of length one.



- (2) For each line segment, mark the midpoint



- (3) For each line segment with a marked midpoint, flip a coin and move the midpoint either vertically up (heads) or vertically down (tails) by $1/4$ th of the horizontal length of the line segment.



- (4) Repeat at (2). Stop when you have looped N times through steps (2)–(4).

- A. With your group, a piece of gridded paper, and a coin, run the 3-level Coastline Algorithm and sketch the result. Compare your result with other groups.
- B. Do you understand the N -level Coastline Algorithm? Answer these questions.
- How many line segments will there be in the final coastline?
 - For each dot (endpoint of a line segment) in the final coastline, what is the x -coordinate? (*The y coordinate is only determined by running the algorithm, but you can program the x -coordinates any time you need them.*)
- C. To turn the Coastline Algorithm into a MATLAB/OCTAVE code you will need to answer these questions, among others. (*Answers will vary depending on your code design.*)
- What MATLAB function will you use to “flip a coin”?
 - How do you store the coastline at each stage?
 - How do you get the y -coordinate of the midpoint of a line segment?
 - How many `for` loops will be in your code?
- D. Write (and run) a MATLAB/OCTAVE code `coastline.m`, either a script or a function.
- E. Though random, you may perceive a coastline from this algorithm as fairly smooth, or at least as not being very interesting. Explain.
- F. Change the algorithm (and the code) so that the coastline is more interesting.