

Skill Problem D: Generating Self-Aware Arrays

An array A of size n is *self-aware* if, for each $i < n$, $A[i]$ is the exact number of occurrences of i in A .

Example:

$$A = [2, 0, 2, 0]$$

($A[0] = 2$ and A contains two occurrences of 0. $A[1] = 0$ and A contains 0 occurrences of 2. $A[2] = 2$ and A contains two occurrences of 2. $A[3] = 0$ and A contains 0 occurrences of 3.)

For this problem, you must do the following:

1. [6 points] Implement the following function.

```
public static List<int[]> allSelfAwareArrays(int n)
```

This function should accept any positive integer n and return a list of all self-aware arrays of length n . For instance, in the return list L for `allSelfAwareArrays`, on input 4, one of the arrays that should belong to L is $[2, 0, 2, 0]$.

You must provide a `main` method that will print to console the list of arrays returned for each positive integer n .

For full credit on this part of the problem, your function must be able to return the full list of all self-aware arrays for each of the inputs 1, 2, 3, ..., 7. Points will be deducted if any of your lists contain arrays that are *not* self-aware.

Note: You will not get full credit if your code cannot be executed for $n = 7$ (if, after several minutes of running, the application does not end, I will assume that you have run out of memory).

2. [3 points] What is the running time of your algorithm? If it is exponential or worse, prove that. If it runs in polynomial time, then give the asymptotic running time and prove your answer.

What to turn in.

- Your implementation in code of the function `allSelfAwareArrays`
- Your document that explains your answer to part 2.