**Recursive Strategy: Print a list of integers BACKWARDS**

|  |  |
| --- | --- |
| **Problem** | Print a list of integers in reverse order, e.g. [ 3, 14, 28, 7]Would print 7, 28, 14, 3 |
| **What is the smallest version of this problem?** *(leads to base case)* | If the array consists of one element, print the only element (which happens to be the 0th element) |
| **What recursion strategy should I use:*** **Forwards recursion***(each recursive step gets larger; the base case is based on # of iterations)*
* **Backwards recursion** 🡨 classical*(each recursive step gets smaller until the base case is reached)*
 | Backwards recursion |
| **For recursive cases, should I**:* **Process first and recur last?** *(process as I move up the recursive stack)*
* **Recur first and process last?** *(process as I move down the recursive stack)*
 | Process as I move **down** the stack.Note that the first time we print will be the base case of [ 7 ]. The second time we print will be the next-to-last case of [ 28, 7 ]. And so on… |
| **What should each recursive step do?** | Print the first integer of the Array |
| **For backwards recursion solutions:****How should the problem be reduced on each step?**  | Send progressively smaller Arrays on each iteration where each Array is missing its first element.\* |
| **For forwards recursion solutions:****How should I keep track of the running answer?** | N/A |

\*Instead of sending smaller and smaller arrays each time, could you also send the ***same array*** to each step as well as the size of the smaller virtual array (which would decrement by one on each recursion)?

(The “virtual array” would consist of the last x elements, where x = the progressively reducing size.)