**Recursive Strategy: Backwards String**

|  |  |
| --- | --- |
| **Problem** | Recursively, print a string backwards.  “Hello World” 🡺 “dlroW olleH” |
| **What is the smallest version of this problem?** *(leads to base case)* | Printing a string of length 1 backwards  “O” 🡺 “O” |
| **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 **up** the stack. |
| **What should each recursive step do?** | Print the last character |
| **For backwards recursion solutions:**  **How should the problem be reduced on each step?** | Send progressively smaller Strings on each iteration where each String is missing its *last* character. |
| **For forwards recursion solutions:**  **How should I keep track of the running answer?** | N/A |

Alternate approach:

|  |  |
| --- | --- |
| **Problem** | Recursively, print a string backwards.  “Hello World” 🡺 “dlroW olleH” |
| **What is the smallest version of this problem?** *(leads to base case)* | Printing a string of length 1 backwards  “O” 🡺 “O” |
| **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. |
| **What should each recursive step do?** | Print the first character |
| **For backwards recursion solutions:**  **How should the problem be reduced on each step?** | Send progressively smaller Strings on each iteration where each String is missing its *first* character. |
| **For forwards recursion solutions:**  **How should I keep track of the running answer?** | N/A |