Homework 6 – Due Apr. 25th 23:59 KST

Instructions: Complete the implementation and turn it in before the due date. Any deviations from the instructed deliverable format will result in a deduction of grade. DO NOT COPY OTHER'S WORKS!

In this assignment, you will again work with NumberSnake, except this time the implementations must be given in recursive form. You don't have to implement everything again, just the following three methods:

- public boolean isValidSnake()
- public String toString()
- public Body getMaxValidSnake(Body[] b)

The last method is a new one, which returns the longest among all new valid snakes that can be composed from the set of bodies 'b'. For example, if b is {(H, 3), (2, 4), (5, 2), (3, 1), (3, T), (2, 1), (1, T)} (with abuse of notation), then it should return a new snake containing (H,3)-(3,1)-(1,T). Notice that (H,3)-(3,T) is also a possible valid snake, but is ignored since the former snake is the longest. If there are multiple longest snakes, you may return any one of them. If no valid snakes can be formed, then return a null.

Restrictions & conditions: No importing of extra packages, no 'trivial' recursion, and no usage of techniques/methods not taught in class. The last restriction is to ensure prevention of unnecessary misunderstandings regarding cheating. You are, however, allowed to have helper methods that do the main recursion.

Once you're done, submit the usual NumberSnake.java file with the above methods implemented recursively. In addition to the

Rubric: Grading will be based on, but not limited to, the following criteria.

- Documentation (30 points): For each of the three main methods, you should provide extensive descriptions in the header comments. In particular, the descriptions should include the outline of your algorithm as well as the clear description of the base and recursive cases.
- Correctness (70 points): Your implementation should behave as specified above in an error-free manner.

Deliverable: A single NumberSnake.java file not part of any package structures.