COS 485: Program #3 – Maximum Run Sum

Objectives: Designing and coding a dynamic programming algorithm

Your input is an array of integer values such as [3, -2, 5, -4, -3, 7, -1, 3, -7, 2].

A "run" in the array is any series of adjacent values such as [3, -2, 5] or [7, -1, 3].

The run [3, -2, 5] has a sum of 6. The run [7, -1, 3] has a sum of 9.

Your task is to design a dynamic programming algorithm to find the run with the maximum sum.

I suggest you try this by hand. Think about how you can use the answers to smaller runs to calculate the answers to longer runs. Faster algorithms and algorithms using less space will receive more credit.

Setting up the project in Eclipse:
Create a new project similar to how you set up program 1:

- It will use MaxRunSumTester.jar, the same Scaffold jar, and starting code MaxRunSum.java
- In the run configuration set **Main class** to: tester/MaxRunSumTester

What to turn in:
Written Report turned in through Brightspace (must be .doc, .docx, or .pdf)

1. A brief English description of your algorithm
2. An analysis of its worst case execution time
3. An analysis of the extra space it uses
4. A screen shot of the report tab
5. A screen shot of your results for test4.txt

Electronic Submit
From a Unix machine in the lab run the program "submit" to submit your files.
Submit your source code (.java files) and compiled code (.class files) to the directory: **prog3**

Grading:

- 10 points – English explanation of your algorithm
- 10 points – clearly explained run time analysis
- 10 points – space analysis
- 10 points – is a dynamic programming algorithm
- 50 points – correct on all test cases
- 10 points – time and space efficient