

---

## Project 2 - Baseball Fun

This project is due **Friday, December 9 at 11:59PM**. Upload a zipped file named `yourlastnameFirstinitial_proj2` to Canvas. The file must include a(n):

1. Executive Summary Report in PDF format
2. Python ( `.py` ) file
3. `readme.txt` file with instructions on how to run your code

---

Consider two baseball teams (*RedSox* and *Yankees*), playing a series of games until one of the teams wins  $n$  games of the series. Assume that the probability of the *RedSox* winning a game is the same for each game and equal to  $p$  and the probability of the *RedSox* losing a game is  $q = 1 - p$  (There are no ties.) Let  $P(i, j)$  be the probability of the *RedSox* winning the series if the *RedSox* need  $i$  more games to win the series and the *Yankees* need  $j$  more games to win the series. The recurrence relation for  $P(i, j)$  is as follows:

$$P(i, j) = pP(i - 1, j) + qP(i, j - 1) \text{ for } i, j > 0$$

with initial conditions:

$$P(0, j) = 1 \text{ for } j > 0 \text{ (Red Sox won)}$$

$$P(i, 0) = 0 \text{ for } i > 0 \text{ (Red Sox loses)}$$

1. Write Python 3.x code for a recursive algorithm that computes the odds of winning a series of  $n$  games in the series. Your program should input the number of games  $n$  (needed to win) and the probability  $p$  of the *RedSox* winning the game. The function returns the probability. **(25 points)**
2. Find the probability of the *RedSox* winning a seven-game series if the probability of it winning a game is 0.4. This means that  $n$  is 4. In your report, show your dynamic programming table  $P(i, j)$  with its entries rounded-off to two decimal places. **(20 points)**
3. Write Python 3.x code of the dynamic programming algorithm that computes the odds of winning a series. Your program should input the number of games  $n$  needed to win and the probability  $p$  of the *RedSox* winning the game. The function returns the probability. **(25 points)**
4. Describe the pros and cons of both your algorithms in your report. **(10 points)**

5. Use a similar infrastructure that you used with Project 1 to time and test your algorithms. Record your algorithms time and space efficiencies as well as your test runs in your report. **(10 points)**
6. Write a professional report. **(10 points)**

**Before handing in your assignment, be sure to check the following:**

1. Implementation
  - (a) Execution. The code runs properly and is written in logical and understandable format.
  - (b) Heavily Documented
    - i. A header block that contains your name, assignment, brief description of the code
    - ii. Comments throughout the code to help the grader understand your thought process
    - iii. Your code comes with a readme file with instructions on how to run your code.
2. Executive Summary Report. Your report contains the following:
  - (a) A summary of your findings
  - (b) A summary of your time and space efficiency analysis
  - (c) Answers to all above questions
  - (d) Pseudocode for your algorithms
  - (e) Screenshots of the command-line of your algorithm being run
  - (f) Form and Style. Grammatically correct with no spelling errors, easy to read and understand