Algorithms
Dynamic Programming Project (100 pts)
Trebuchet Problem

You are not allowed to use the internet or consult any external references. You may use lecture slides.

1 Problem Description

1.1 Introduction

The trebuchet problem discussed in class and on the supplemental handout is briefly defined as follows: Given t targets at 1-meter intervals and p pumpkins to be thrown via trebuchet, what is the *worst-case* minimum number of throws necessary to determine the maximum distance a pumpkin can be thrown without shattering on impact?

.

1.2 Recurrence Relation

Recall from the handout that the recurrence relation for the trebuchet problem is:

$$T(p,t) = 1 + \min_{1 \le x \le t} \Big(\max \Big[T(p-1, x-1), T(p, t-x) \Big] \Big)$$

$$\text{Base Cases: } \begin{cases} T(p,0) = 0 \\ T(p,1) = 1 \\ T(1,t) = t \end{cases}$$

2 Deliverables

Please submit all of the items requested below in a single PDF file on Canvas.

1. [20] Write a recursive algorithm to solve the problem for $p = 3$, $t = 16$. Don't use any shortcuts (i.e. there should be a for-loop from 1 to t). Including the initial call, how many calls are made to the function? Include your code in the report.
2. [10] Run the code from the previous algorithm on a few combinations of p and t to characterize the growth in runtime as p and/or t increase. Provide a table or plot of your results and discuss.

3. [20] Implement a dynamic programming algorithm (that uses a table to avoid recomputation) to compute the minimum throws necessary. Include code in your report.

4. [10] Run the code from the previous algorithm on a few combinations of p and t to characterize the growth in runtime as p and/or t increase. Provide a table or plot of your results and discuss.

5. [10] Implement a traceback step that identifies which targets are attempted and what the result is (shattered or survived). Recall that the algorithm should be finding the outcome for the *worst-case*. Include code in your report. **For consistency with our solutions, if there are multiple targets that can be attempted which result in the same outcome for the minimum number of throws, choose the target that is closer. Additionally, if at that target, the minimum number of throws is the same regardless of if the pumpkin shatters or survives, opt for it to shatter.**

6. [20] Demonstrate that your code works correctly by showing its results on the following instance:

   $p = 5$, $t = 100$.

   Output Format
   The output consists of two lines:

   - The first line prints minimum number of throws necessary to find the maximum distance.
   - The second line prints the targets attempted in the order in which they were attempted. If the pumpkin shatters at a target, make the target number negative.

   For the example $p = 2$, $t = 4$, the output could look like:

   ```
   3
   1 3 4
   ```

   While there is only one correct solution of 3 necessary throws, there are multiple ways to get there. Use the guidelines in Deliverable #5 in deciding which targets to attempt.

7. **Verification: you don't need to turn in this part, just do the Canvas quiz.**

   [10] Demonstrate that your code works for larger values of p and t by correctly showing the results for the input on the Canvas quiz.