ECE 374 B ✧ Spring 2024
Homework 6

- **Submit your solutions electronically on the course Gradescope site as PDF files.** please use the LaTeX solution template on the course web site. If you must submit scanned handwritten solutions, please use a black pen on blank white paper and a high-quality scanner app (or an actual scanner, not just a phone camera).

☞ Some important course policies ☜

- **You may use any source at your disposal**—paper, electronic, or human—but you ***must*** cite ***every*** source that you use, and you ***must*** write everything yourself in your own words. See the academic integrity policies on the course web site for more details.
- **Avoid the Three Deadly Sins!** Any homework or exam solution that breaks any of the following rules will be given an ***automatic zero***, unless the solution is otherwise perfect. Yes, we really mean it. We're not trying to be scary or petty (Honest!), but we do want to break a few common bad habits that seriously impede mastery of the course material.
- For algorithmic problems, **we will be grading clarity and conciseness of solutions** (in addition to correctness) which admittedly is a subjective measure, but it is necessary. Being able to communicate your code is a far more important skill than being able to code itself.
- Solutions to a dynamic programming problem have (at minimum) three things:
  - A recurrence relation
  - A *brief* description of what your recurrence function represents and what each case represents.
  - A *brief* description of the memory element/storage and how it's filled in.
- Last minute tips:
  - Always give complete solutions, not just examples.
  - Always declare all your variables, in English. In particular, always describe the specific problem your algorithm is supposed to solve.
  - Never use weak induction.

See the course web site for more information.

If you have any questions about these policies,
please don't hesitate to ask in class, in office hours, or on Piazza.

1. **Largest Square of 1's** You are given a $n \times n$ bitonic array A. Describe and analyze a dynamic programming algorithm that returns the side length of the largest square of 1's.

$$\begin{array}{c|c}
 & j \rightarrow \\
\hline
i \downarrow & \begin{bmatrix}
1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 & 1 & 0 \\
0 & 0 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 \\
1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 1 \\
1 & 1 & 1 & 1 & 0 & 0 & 1 & 1 & 0 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 0 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 \\
1 & 1 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 0 \\
1 & 1 & 0 & 0 & 0 & 1 & 1 & 1 & 1 & 1 \\
1 & 0 & 0 & 0 & 1 & 0 & 1 & 1 & 1 & 0 \\
0 & 1 & 1 & 1 & 1 & 0 & 1 & 1 & 1 & 0
\end{bmatrix}
\end{array}$$

Figure 1: Example: The output is the side-length of the largest square of 1's (4 in the case of the graph above, yes there can be multiple squares of the greatest size).

2. The traditional world chess championship is a match of 24 games. Each game ends in a win, loss, or draw (tie) where a win counts as 1 point, a loss as 0 point, and a draw as 1/2 point. The current champion retains the title in case he scores 12 or above. The players take turns playing white and black. In the first game, the champion plays white. The champion has probabilities w_w, w_d, and w_l of winning, drawing, and losing playing white, and has probabilities b_w, b_d, and b_l of winning, drawing, and losing playing black.

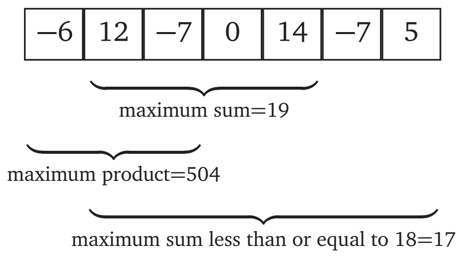
   (a) Write a recurrence for the probability that the champion retains the title.
   hint : Let g and i be the two parameters, where g is the number of games left to play and i is the amount of points remaining for the champion to retain title(which may be a multiple of 1/2).

   (b) Based on your recurrence, give a dynamic programming algorithm to calculate the champion's probability of retaining the title.

   (c) Analyze the runtime of the algorithm in (b) for an n game match.

3. Plum blossom poles are a Kung Fu training technique, consisting of n large posts partially sunk into the ground, with each pole pi at position (xi, yi). Students practice martial arts techniques by stepping from the top of one pole to the top of another pole. In order to keep balance, each step must be more than d meters but less than 2d meters. Give an efficient algorithm to find a safe path from pole ps to pt if it exists.

4. Suppose you are given an array $A[1..n]$ of arbitrary real numbers. Recall a *subarray of an array A* is by definition a *contiguous* subsequence of A. Define the sum and product of an empty array to be 0 and 1, respectively. For any array $A[i..j]$ where $i \leq j$, define its sum and product to be
$$\sum_{k=i}^{j} A[k] \quad \text{and} \quad \prod_{k=i}^{j} A[k],$$
respectively. For the sake of analysis, assume that comparing, adding and multiplying any pair of numbers takes $O(1)$ time.



   (a) Describe and analyze an algorithm to compute the *maximum sum* of any subarray of A. For example, given $A = [-6, 12, -7, 0, 14, -7, 5]$, your algorithm should return 19 as illustrated above.

   (b) Describe and analyze an algorithm to compute the *maximum product* of any subarray of $A[1..n]$. For example, given $A = [-6, 12, -7, 0, 14, -7, 5]$, your algorithm should return 504 as illustrated above.