Notes on Grading

- You can write “I don’t know” for any question and receive 25% credit. You can take this option for any numbered problem, but not for part of a problem. For example, you can answer 3.1 and write “I don’t know” for 3.2, but you can’t write part of the solution for 3.2 and then write “I don’t know” for the rest.

- You get a 10% bonus for typing your homework. You are encouraged to use \LaTeX. You must type your entire homework to receive the bonus. The 10% bonus does not apply to problems answered with “I don’t know.”

Important Note

All the problems on this homework ask for dynamic programs.

To answer a problem:

1. Briefly explain the recursive substructure of the problem.
2. Give the rule for filling in the dynamic programming table.
3. Give the asymptotic running time for the dynamic program.
4. Give the asymptotic space required for the dynamic program.

Then pick one of the problems and write out a complete implementation (in pseudo-code) for that algorithm.
1 Fairly splitting an array

Given an array $A$ of positive integers and an integer $k$, devise a dynamic program to split it into $k$ arrays, $A_1, \ldots, A_k$, as “fairly” as possible. Note that each $A_i$ must be a contiguous piece of $A$.

For this problem, a split would be perfectly fair if every array $A_i$ had the same sum, i.e. if for all $i$

$$\sum_j A_i[j] = C$$

where $C = \sum_j A[j]/k$.

Since a perfectly fair solution may not always be possible, your goal is to split $A$ so that

$$\max_i \left| C - \sum_j A_i[j] \right|$$

is minimized.

2 Breaking a string into words

Suppose you are given a string $s$ of letters with no spaces, and you would like to break it into English words, or at least things that look like they might be words.

You have a function $q$ that takes a string as input and tells you a score indicating how much the string looks like an English word. Higher scores are more English-like.

Devise a dynamic program for breaking $s$ into words $w_1, \ldots, w_k$, such that $\sum_i q(w_i)$ is maximized. Note that $k$ is not a parameter of the problem – you can break $s$ into any number of words.

3 Finding big squares

Show a dynamic program that, given an $n \times m$ matrix $A$ of 0s and 1s, finds the largest square region of $A$ that is all 1s.

4 Matched parentheses

A string $s$ of parentheses (“(“ and “)”) and brackets (“[“ and “]”) is matched if it is of one of the following forms:
• the empty string
• $(x)$, where $x$ is matched
• $[x]$, where $x$ is matched
• $xy$ where $x$ and $y$ are both matched.

Describe a dynamic program to determine whether a string $s$ is matched.

5 Correcting typos

Give a dynamic program that, given a string of parentheses and brackets, computes the edit distance to the nearest matched string.