CS361 Dynamic Programming Guide

Matthew Rackley

February 2024

How to format your Dynamic Program

$$Function(n) = \begin{cases} \text{Base Case(s)} & Conditional(s) \\ \text{Outside of the Base Cases} & Conditional \\ \text{Decision Making Function(s)} & O.W. \end{cases}$$

To write the stepwise function in LaTeX, use the ‘cases‘ environment provided by the ‘amsmath‘ package. Here’s an example:

```
\[
f(n) =
\begin{cases}
2n & \text{if } n = 2, 4, 8 \\
-\infty & \text{if } n < 8 \\
1 + \max(f(n-2), f(n-4), f(n-8)) & O.W.
\end{cases}
\]
```

In this example, we define the function $f(n)$ using different cases based on the value of n.

$$f(n) = \begin{cases} 2n & \text{if } n = 2, 4, 8 \\ -\infty & \text{if } n < 8 \\ 1 + Max(f(n-2), f(n-4), f(n-8)) & O.W. \end{cases}$$

It’s important to recognize that the standard for evaluation in this format is from the top down. So make sure your terminal cases are evaluated first, before your recursive functions!

Be aware you can always enhance your dynamic program with variables and data structures to get what you need out of the program. You may assume certain global vars exist to make it easier.

A key observation is if you minimize the number of args in $f(n)$, the number of args *very closely* related to your expected running time for your bottom up solution.