# AD Exam 2019
# Dynamic Programming

Noah Stonall [mjh831]

March 2019

## Disposition

- Dynamic Programming
  - Divide the problem into subproblem
  - Solve the subproblems only once and save the solution
  - Top-down memoization: recursion
  - Bottom-up: 2 nestede `for`-loops
- Longest common subsequence
  - Finds a string that both the initial strings have in common
  - Demonstration by figure
  - Runtime $\Theta(mn)$ - 2 nested `for`-loops
  - Correctness - loop invariant
- Perspective - Greedy algorithms
  - Optimises in a different way

## Answer