# Week 5

Kaung Khant Lin
6540131
542

# Edit Distance

1.

| Condition | Edit Operation | Next State |
|---|---|---|
| A[i] == B[j] | None | (i + 1, j + 1) |
| A[i] != B[j] | Insert B[j] in front of A[i] | (i , j + 1) |
| A[i] != B[j] | Delete A[i] | (i+1 , j) |
| A[i] != B[j] | Change A[i] to B[j] | (i + 1, j + 1) |

2. The beginning state is We start at the very first letter of both strings (0, 0)
3. The cost is the number of remaining letters in B: len(B) - j.
4. The cost is the number of remaining letters in A: len(A) - i.

# Minimum Coin Change (Dynamic Programming)

1. minicoin(v1) (Top-Down): Calculates the largest amount (v1) first. The smaller amount (v2) is calculated during v1's execution.
2. minicoin(v2): The minicoin(v1) calculation pauses, waiting for minicoin(v2) to finish and return its result.
3. mm[v2]: Since minicoin(v2) finishes before minicoin(v1), its result is stored in the memory array (mm) first.