

# Assignment 9

csci2200, Algorithms

**Honor code:** *Work on this assignment alone or with one partner. Between different teams, collaboration is at level 1 [verbal collaboration only]. There are lots of resources online, such as animations, visualizations, practice problems, videos, and solutions— which you are encouraged to explore to deepen your understanding. However, you must be careful not to search for the specific problems in the assignment with the intent of getting hints for the solution. Searching for the assignment problems on the internet violates academic honesty for this class.*

---

**String shuffling:** A *shuffle* of two strings  $A$  and  $B$  is formed by interspersing the characters into a new string, keeping the characters from  $A$  and  $B$  in the same order.

For example, the string BANANAANANAS is a shuffle of the string BANANA and ANANAS (in several different ways, actually: BANANAANANAS, BANANAANANAS and also BANANAANANAS).

Similarly, the strings ANEVGARIN and ANEGAVRIN are both shuffles of NEVER and AGAIN.

**The problem:** Given three strings  $A[1..m]$ ,  $B[1..n]$  and  $C[1..m+n]$ , come up with an efficient algorithm to determine whether  $C$  is a shuffle of  $A$  and  $B$ .

1. Define your subproblem. Clearly state what function you will compute, what value it should return, what the arguments represent.
2. Argue optimal substructure and give a recursive definition of the subproblem (which you defined above). Without DP and memoization, what would the worst-case running time be?
3. Give a recursive, top-down dynamic programming approach with memoization. What is the (worst-case) running time?

Include at least 10 test cases to show that your algorithm works. Here are a few suggested ones:

BANANA, ANANAS, BANANAANANAS : True

AA, BA, AABA : True

BA, AA, AABA : True

A, BA, AAB : False.

*We expect: A Python notebook with answers to all questions. For this assignment you will turn in only the notebook, nothing else.*