CSE525 Monsoon 2020 — Ashima Garg (PhD19003)
Homework 1 Problem 1 — Pramil Panjawani (PhD19008)

You CANNOT consult any other person or online resource for solving the homework problems. You can definitely ask the instructor or TAs for hints and you are encourage to do so (in fact, you will get useful hints if you ask for help at least 1-2 days before the due date). If we find you guilty of academic dishonesty, penalty will be imposed as per institute guidelines.

(a) Write a recursive algorithm to merge two sorted arrays of equal size.

Solution: ⟨⟨*The explanation is added in the box below.*⟩⟩

```
⟨⟨A and B are sorted⟩⟩
⟨⟨C is for storing the result after merging⟩⟩
def Merge(A[1...n], B[1...n], C[1...2n]):
  return MergeRecursive(A, B, C, n, n, 2n)

⟨⟨Recursively merges A[1...i] and B[1...j] and puts the result in C⟩⟩
def MergeRecursive(A, B, C, i, j, k):
  if(i == 0): ⟨⟨Check if A is empty⟩⟩
      C[k] = B[j]
      MergeRecursive(A, B, C, i, j − 1, k − 1)
  else if(j == 0): ⟨⟨Check if B is empty⟩⟩
      C[k] = A[i]
      MergeRecursive(A, B, C, i − 1, j, k − 1)
  else: ⟨⟨Check if neither is empty⟩⟩
      if (A[i] < B[j]):
          C[k] = B[j]
          MergeRecursive(A, B, C, i, j − 1, k − 1)
      else:
          C[k] = A[i]:
          MergeRecursive(A, B, C, i − 1, j, k − 1)
```

Explanation

Step 1: Take in the sorted arrays(A,B) of size n.
Step 2: Take an empty array C of size 2n.
Step 3: First check for base condition if A or B is empty.
Step 4: If yes put the elements of other directly at C[k] and reduce the index of C.
Step 5: If neither is empty then compare the elements of A and B.
Step 6: If A[i] is greater then put that element at C[k] and decrement i and k
else put B[j] at C[k] and decrement j and k.
Step 7: Repeat till both A and B are empty

■

(b) Write the time complexity of your recursive merge algorithm.

Solution: Let $T(i,j)$ denotes the worst-case time taken by MergeRecursive to merge two sorted arrays of length i and another sorted array of length j. The recurrence relation for $T(i,j)$ is given below.

$$T(i,j) = O(1) + max\{T(i-1,j) + T(i,j-1)\}$$
$$T(0,j) = O(1) + T(0,j-1) = O(j) \ldots \text{(base condition i)}$$
$$T(i,0) = T(i-1,0) + O(1) = O(i) \ldots \text{(another base condition ii)}$$

The time complexity of Merge on two n-sized sorted array is given by $T(n,n)$. The solution of the recurrence is given below.

$$T(n,n) = max\{T(n-1,n), T(n,n-1)\} + O(1)$$
$$Since, T(n-1,n) = T(n,n-1)$$
$$T(n,n) = T(n-2,n) + O(1) + O(1)$$
$$= T(n-3,n) + O(1) + O(1) + O(1)$$
$$= T(n-k,n) + O(k)$$
$$if\, k = n,$$
$$T(0,n) = O(n) \ldots \text{(from base condition i)}$$
$$= O(n)$$

■