TODO-Your Group Number and Names Here due: 20 September 2019

**CSCI 532 Problem 2-1**

Collaborators: *TODO-list your collaborators here*

Give a linear-time algorithm that takes two sorted arrays of real numbers as input, and returns a merged list of sorted numbers. You should give your answer in pseudocode. Your answer should contain:

- A prose explanation of the algorithm.

  **Compare the first item of Array1 and Array2, if the item of Array1 is less than the item of Array2, add this item to the MergedArray and remove it from Array1. If the element is not less than the element of Array2, instead, add the item of Array2 to the MergedArray and remove it from Array2. Do this until both arrays are empty. To save some time, if either array becomes empty, append the other array to the end of the MergedArray.**

- Psuedocode. (Be sure to review the two resources on pseudocode that were posted as readings for Week 2! I also suggest the algorithm / algorithmx package in LaTex.)

**Algorithm** Merge two sorted arrays of real numbers into a list of sorted real numbers

```
procedure Merge2SortedArrays(inArr1, inArr2)    ▷ This procedure assumes that inArr1 and inArr2 are sorted in ascending order
    Arr1 = inArr1
    Arr2 = inArr2
    mergedArr = empty array of size |arr1| + |arr2|    ▷ This will be the sorted merged array
    while |arr1| > 0 && |arr2_idx| > 0 do    ▷ While the arrays are not empty
        if arr1[0] < arr2[0] then
            mergedArr.append(arr1[0])    ▷ Add the first element of arr1 to the merged array
            arr1.remove(0)    ▷ Remove the element from arr1
            if arr1 == [ ] then    ▷ If arr1 is empty after removing an element
                mergedArr.append(arr2)    ▷ Append the entire arr2 to merged array
                arr2 = [ ]
            end if
        else
            mergedArr.append(arr2[0])
            arr2.remove(0)    ▷ Remove the element from arr2
            if arr2 == [ ] then    ▷ If arr2 is empty after removing an element
                mergedArr.append(arr1)    ▷ Append the entire arr1 to merged array
                arr1 = [ ]
            end if
        end if
    end while
end procedure
```

- The decrementing function for any loop or recursion.
  - **while** $|arr1| > 0$ && $|arr2\_idx| > 0$ **do**

    $D(x)$ where $x = 2(|arr1| + |arr2|) - |mergedArr|$

- Justification of why the runtime is linear.

  **The algorithm is O(n + m) where n and m are the lengths of inArr1 and inArr2 respectively. It is O(n + m) because for each iteration in the while loop, an item is appended to the MergedArr and removed from its own array. Thus, for each iteration, the length of one of the two input arrays is decreasing and the length of the merged array is increasing. Therefore, the algorithm cannot be greater than O(n + m). Technically, in expectation, it is less than O(n+m) because if one array becomes empty, it appends the rest of the remaining array to MergedArr**

| | |
|---|---|
| TODO-Your Group Number and Names Here | due: 20 September 2019 |

**CSCI 532 Problem 2-2**

Collaborators: *TODO-list your collaborators here*

EPI-C++ 15.4 / EIP-Java 15.5 (Generate the Power Set) gives code to compute the power set of a set (without duplicates). Present this problem and solution in your own words using pseudocode.

TODO-Your Group Number and Names Here due: 20 September 2019

**CSCI 532 Problem 2-3**

Collaborators: *TODO-list your collaborators here*

In EPI 15.1 (The Towers of Hanoi Problem), prove that the algorithm as presented terminates. In particular, you should give the decrementing function for the recursion.

TODO-Your Group Number and Names Here due: 20 September 2019

**CSCI 532 Problem 2-4**

Collaborators: *TODO-list your collaborators here*

For the stock market problem discussed in class on September 6th (and in CLRS 4.1), walk through the algorithm for the following input:

$$\texttt{price} = \{3, 6, 8, 2, 1, 10, 5, 7\}.$$

| | |
|---|---|
| TODO-Your Group Number and Names Here | due: 20 September 2019 |

**CSCI 532 Problem 2-5**

Collaborators: *TODO-list your collaborators here*

Prove using induction that the closed form of:

$$T(n) = \begin{cases} 1 & n = 1 \\ T(n-1) + n & n > 1 \end{cases}$$

is $O(n^2)$.

TODO-Your Group Number and Names Here due: 20 September 2019

**CSCI 532 Problem 2-6**

Collaborators: *TODO-list your collaborators here*

What is the closed form of the following recurrence relations? Use Master's theorem to justify your answers:

1. $T(n) = 16T(n/4) + \Theta(n)$
2. $T(n) = 2T(n/2) + n \log n$
3. $T(n) = 6T(n/3) + n^2 \log n$
4. $T(n) = 4T(n/2) + n^2$
5. $T(n) = 9T(n/3) + n$

Note: we assume that $T(1) = \Theta(1)$ whenever it is not explicitly given.

TODO-Your Group Number and Names Here due: 20 September 2019

**CSCI 532 Problem 2-7**

Collaborators: *TODO-list your collaborators here*

*The skyline problem:* You are waiting for the ferry across the river to get into a big city, and notice $n$ buildings in front of you. You take a photo, and notice that each building has the silhouette of a rectangle. Suppose you represent each building as a triple $(x_1, x_2, y)$, where the building can be seen from $x_1$ to $x_2$ horizontally and has a height of $y$. Let `rect(b)` be the set of points inside this rectangle (including the boundary). Let `building` be the set of $n$ triples. Design an algorithm that takes `buildings` as input, and returns the skyline, where the skyline is a sequence of $(x, y)$ coordinates defining $\cup_{b \in \texttt{buildings}} \texttt{rect}(b)$.

TODO-Your Group Number and Names Here due: 20 September 2019

**CSCI 532 Problem 2-8**

Collaborators: *TODO-list your collaborators here*

The `rand()` function in the standard C library returns a uniformly random number in `[0,RANDMAX-1]`. Does `rand`() $\bmod n$ generate a number uniformly distributed in $[0, n-1]$?

Note I: This is the second variant in EPI 5.12.

Note II: When asked questions of this form, you are expected to justify your answer.

| | |
|---|---|
| TODO-Your Group Number and Names Here | due: 20 September 2019 |

**CSCI 532 Problem 2-9**

Collaborators: *TODO-list your collaborators here*

Algorithms where we use randomization to find a deterministic answer are known as Las Vegas algorithms. Monte Carlo algorithms also use randomization, but might not always give the right answer; however, they either have a high probability of being correct or close to correct.

(a) Give a Monte Carol algorithm to estimate $\pi$.

(b) Let $n$ be the number of random numbers used by your algorithm. Explain why as $n \to \infty$, the expectation of the output for your algorithm is $\pi$.

(c) Implement this algorithm and plot a line graph of the values returned for at least 10 values of $n$.

Note: We can use the function `randReal`$(a, b)$ that returns a random real number between $a$ and $b$ inclusive.