1. **Parallel Prefix Sum**: Given input array [8,9,6,3,2,5,7,2], output an array such that each output[i]=sum(array[0],array[1],...array[i]), using the Parallel Prefix Sum algorithm from lecture. Show the intermediate steps. Draw the input & output arrays, and for each step, show the tree of recursive task objects that would be created (where a node’s child is for two problems of half the size) and the fields each node needs. Do not use a sequential cut-off.

2. **Parallel Prefix FindMin**: Given input array [8,9,6,3,2,5,7,4], output an array such that each output[i]=min(array[0],array[1],...array[i]). Show all steps, as above.

3. Show that Quicksort with sequential partitioning, but parallel recursive sorting, is indeed O(n), by solving the recurrence relation shown in lecture: T(n) = n + T(n/2)

4. Show that a completely parallel Quicksort, with parallel partition and recursion, is O(log² n), by solving the recurrence relation shown in lecture: T(n) = O(log n) + T(n/2)