

Just one problem for your first assignment.

General homework policy: You can look up material on the web and books, but you cannot look up solutions to the given problems. You can work in groups, but must write up the answers individually. You can either typeset it or write it by hand. If you write it by hand it must be readable by the Professor and TA to be graded.

Problem 1: Parallel Merging

Describe a divide-and-conquer algorithm for merging two sorted arrays of lengths n into a sorted array of length $2n$. It needs to run in $O(n)$ work and $O(\log^2 n)$ depth. As defined in the assigned reading, *work* is the total number of operations executed by a computation, and *depth* is the longest chain of sequential dependencies in the computation.

You can write the pseudocode for your algorithm so that it looks like your favorite sequential language (C, Java, ML, ...), but with an indication of which loops or function calls happen in parallel. For example, use `parallel for` for a parallel for loop, and something like:

```
parallel {  
    foo(x,y);  
    bar(x,y)  
}
```

to indicate that `foo` and `bar` are called in parallel.

You should write down recurrences for analyzing the work and depth for your algorithm and solve them.

You should also prove correctness at the level expected in an algorithms class (e.g. 15-451).