Homework 4: Divide & Conquer

Due: February, 18, 2026 at 11:59 pm

Problem 1. The recurrence $T_A(n) = 7T_A(n/2) + n^2$ describes the running time of an algorithm A. A competing algorithm B has a running time of $T_B(n) = cT_B(n/4) + n^2$. What is the largest integer value for c such that B is asymptotically faster than A?

Solution. $\square$

Problem 2. The Euclidean algorithm is a method for computing the greatest common divisor (gcd) of two numbers, taking advantage of the fact

$$\gcd(a, b) = \gcd(b, a - b)$$

for positive integers a, b satisfying $a \geq b$. Consider the following variation:

$$\gcd(a, b) = \begin{cases} 2\gcd(a/2, b/2) & \text{if } a, b \text{ even} \\ \gcd(a, b/2) & \text{if } a \text{ odd } b \text{ even} \\ \gcd((a-b)/2, b) & \text{if } a, b \text{ odd} \end{cases}$$

a. Prove the variation is correct.

b. Provide an algorithm that uses the variation to compute the greatest common divisor of two numbers a, b in $O(\log(ab))$. Prove the correctness of your algorithm and provide a proof that its worst-case running time is $O(\log(ab))$.

Solution. $\square$

Problem 3. While playing with your pile of n rocks, you begin to wonder about the types of rocks you have. A type of rock is *overly common* if more than half of the rocks you have are of this type.

Unfortunately, you don't have the rock expertise to exactly tell what type a rock is, however, you do have access to a magic machine (i.e. a predicate) that is able to determine if two rocks share the same type.

(a) Describe an algorithm to determine if a collection of n rocks contains an overly common type of rock. If it does, return all of the overly common rocks. Your algorithm should make at most $O(n \log n)$ calls to the magic machine.

(b) Prove the correctness of your algorithm.

(c) Analyze the runtime of your algorithm, justifying that your algorithm makes at most $O(n \log n)$ class to the machine.

Solution. □