CS41 Lab 13: Unconditional Lower Bounds

Thursday, April 23

1. **Almost-Sorting**. Say that an array $A[1 \dots n]$ is *almost sorted* if for all $1 \leq i \leq n$, the ith smallest element in A is either in position $i-1, i$, or $i+1$. Use an encoding argument to show that any comparison-based algorithm that almost-sorts an array requires $\Omega(n \log n)$ comparisons.

2. **The Hiking Problem**. In this exercise, you'll develop better lower bounds for the Hiking Problem. Recall that in this problem, you'd like to meet up with your friend on the Appalachian Trail, but you don't know where your friend is. More formally, your friend is exactly m miles away, but you do not know m in advance, nor do you know in which direction your friend is.

   - We argued in class that even if we knew our friend was, say, m miles north, we'd still need to walk m miles. Now, suppose you know your friend is exactly m miles away, but you don't know which direction. How many miles do you need to travel in the worst case, *in comparison to how far away your friend is*?
   - Extend the argument above to get a $4m$ lower bound on the distance traveled.

3. **Communication Complexity of EQUALITY**. In randomized communication complexity, Alice and Bob have access to a common source of randomness, and may use randomness to help them decide what messages to send. Adding randomness to a protocol can occasionally cause errors, but as long as the protocol outputs the correct answer most of the time (say, with probability $\geq 2/3$) it might be worth the error.

   (a) Give a three-bit randomized protocol for EQ. For any input (x, y), your protocol should compute $EQ(x, y)$ with probability at least $2/3$.

   (b) Give an eight-bit randomized protocol for EQ that has error at most 0.01.