9th tutorial

1. Comparator. Construct a circuit of $\log n$ depth which compares two n-bit numbers x, y, i.e., returns 1 if $x < y$ and 0 otherwise. (This is an implementation of a comparator over the boolean alphabet.)

Hint: recall a task from the previous tutorial...

2. Maximum: Design a comparator network for finding the maximum: given n elements, return a permutation of elements where the last value is the largest one, and the remaining are in arbitrary order.

(A *comparator* is a 2-input, 2-output gate, which for inputs a, b returns $\min(a, b), \max(a, b)$.)

3. Maximum lower bound: Show that to determine the maximum of n numbers, $\Omega(\log n)$ layers of comparators are needed, and in total one needs $\Omega(n)$ comparators.

4. Adjacency. Construct a circuit of $\mathcal{O}(\log^2 n)$ depth which, given the adjacency matrix of G, decides if G is connected.

---

5. Lower Bound. Show that sorting n real numbers $x_1, \ldots, x_n$ can be reduced to computing the (ordered) convex hull of some set of n points in $\mathbb{R}^2$, thus computing the convex hull must take time $\Omega(n \log n)$.

6. Furthest Point. Recall that the ℓ_2 distance of points (a_1, a_2) and (b_1, b_2) is $\sqrt{(a_1 - b_1)^2 + (a_2 - b_2)^2}$. Given a set of points $p_1, \ldots, p_n \in \mathbb{R}^2$, find the two which are farthest from each other.

7. "Furthest" numbers. Given n real numbers $a_1, \ldots, a_n \in \mathbb{R}$, find i, j such that $(a_i - a_j)^2 + (i - j)^2$ is maximized. This is not too hard in $\mathcal{O}(n \log n)$, and a clever trick can get $\mathcal{O}(n)$.