- The problems here are for your practice for the final. The first problem is from Pawel.
- The solutions will be out on the 20th of July.

1. Consider a partially persistent array implementing two operations: lookup(t, a), which returns the content of cell a at time t, and update(a, v), which creates a new version of the whole array, where cell a contains v. We want an implementation which uses space $O(s(m))$ for a sequence of m update operations and takes time $O(q(m))$ to answer lookup(t, a).

   Show that using such implementation we can solve static predecessor search for a set of n integers from $[1, n^2]$ in space $O(n + s(n))$ and time $O(1 + q(n))$.

   Hint. Split the universe into n blocks of length n. Explicitly store the predecessor of the first number in each block. Construct a partially persistent array from which the predecessor of any x can be read, as long as it belongs to the same block.

2. **Pointer machine model** Let $a, b \in \mathbb{R}^+$. $G = (V, E)$ is (a, b)-effective for the range reporting problem if for any query range q, $|W(q)| \leq a(|P \cap q| + \log^b n)$. Let $S = \{q_1, \cdots, q_s\}$ be a set of queries, and let $\alpha > 0$. S is α-favorable if, for each $i, j (i \neq j)$,

   - $|P \cap q_i| \geq \log^\alpha n$.
   - $|P \cap q_i \cap q_j| \leq 1$.

   Read the proof of the following theorem from the paper by Chazelle, and reproduce it in your own words.

   Theorem: If the data structure is (a, b)-effective and the set of queries is b-favorable, then

   $$|V| > |S| \frac{\log^b n}{2^{16a+4}},$$

   for n large enough.

3. **Output sensitive lower bounds** The running time of many problems can be analyzed in an output sensitive way. For example,

   - In range reporting, one is ideally looking for a run time of $O(\text{poly}(\log n) + k)$, where k is the number of points reported.
   - The best algorithm that computes the convex hull of a set of n points run in time $n \log k$, where k is the number of points on the boundary of the convex hull (k could be much smaller than n).

   Consider a set S of n real-valued members, associated with each member one of two possible types. Let $S^{'}$ be the set of n members of one type, and $S^{''}$ be the set of $n^{''}$ members of the other type ($n = n^{'} + n^{''}$). We define a multi-partitioning of S to be a sequence of the members of S such that if $x, y \in S$ have different types and $x < y$, x precedes y in this sequence; this results in partitioning S into alternating blocks of maximal subsequences of members with the same type. In other words, for any two consecutive blocks, the members of one block are of one type, and the members of the other block are of the other type. Without loss of generality, we assume that if members of different types have the same value, the members of $S^{'}$ appear before those of $S^{''}$.

Prove a lower bound of $\Omega(n \log n - \sum_i n_i \log n_i + n)$ on this problem, where n_i is the count of the members of the ith block.

4. For the same problem above, what is the best lower bound you can get in the external memory model? Assume that $n^{'}$ and $n^{'}$ are both multiples of B.

5. **Batched predecessor problem** Given a sorted array Y of length y, one is allowed to preprocess Y and build a data structure in order to answer the following query: Given a sorted array X of length x, $x < y$, find the predecessor in Y of every element in X.

   - In the comparison model, prove a lower bound of $\Omega(x \log(y/x) + x)$ on the batched predecessor problem.
   - Describe an algorithm that runs in the same running time as above.
   - Now consider the problem in external memory. Prove a lower bound of $\Omega\left(\frac{x}{B} \log \frac{y}{x} + x\right)$.