# Data mining: homework 1

Edo liberty

1. Describe an algorithm which samples roughly $n$ elements from a stream, each uniformly and independently at random. More precisely, in every point in the stream, after processing $N$ elements, each of the elements $a_1, \ldots, a_N$ should appear in your sample with probability exactly $n/N$. Note that you clearly do not know $N$ in advance and that you cannot reread the stream.
2. Prove the correctness of your algorithm.
3. What is the complexity of processing each element in your algorithm?
4. Bound the probability that your algorithm samples more than $2n$ elements.

Clarifications:

- The homework should be either handwritten (or printed) and brought to class! To be extra clear, it needs to be on paper. Email submission would be accepted in special circumstances only.
- The elements should be sampled **independently**. Knowing whether or not other elements were sampled should not change the probability of any other elements being sampled.
- That means that Reservoir sampling in NOT the answer! Clearly, in Reservoir sampling if I know that the first $n$ elements are in the output, I can be certain that the $(n+1)$'th element is not. That means that they are dependent!
- The complexity of processing an element can be amortized, if you so choose. That means, the total computation on the entire stream divided by $N$.
- $N$ can be assumed to be larger or equal to $n$.
- $N$ is not known in advance! The algorithm receives $n$ and access to a stream from which it can read elements one by one. In some point the stream suddenly ends. This happens after $N$ elements were read. In this point the algorithm should output its random sample.
- Try to think about the memory footprint of your algorithm as well.