Tutorial 6

1. Bloom filter (see: `http://en.wikipedia.org/wiki/Bloom_filter`) is a probabilistic datastructure for storing sets of items. It is probabilistic in the following sense: If the datastructure says an item is not in the set, the result is certain. However, the data structure will answer with "maybe" for all the items in the set and also some other additional items, in case of false positives. Bloom filters are heavily used by many cloud datastores such as Google Bigtable and Apache HBase as extremely memory efficient caches that avoid disk accesses in case the Bloom filter cache says that the data item does not exist.

   a) Consider a Bloom filter with $k = 2$ hash functions consisting of a bitarray of 1 megabyte of memory. Consider the case where we have inserted already $n = 100000$ unique items to the Bloom filter in question. Now test the membership of an element that has not been inserted in the Bloom filter. What is an approximate probability of a false positive, that is the possibility that the Bloom filter says "maybe" for this item for which it should say "no"?

   b) What is an approximate false positive probability after in total $n = 1000000$ unique elements have been inserted to the same Bloom filter?

2. a) Consider the case of a Bloom filter with 1 megabyte of memory, where we would like to insert at most $n = 1000000$ items. What is the approximate optimal number of hash functions k to minimize the number of false positives? What is the false positive propability with using that k after having inserted n unique items?

   b) What is the approximate optimal number of hash functions k, if we only have 512 kilobytes of memory to store the same $n = 1000000$ items? What is the false positive propability with using that k after having inserted n unique items?