582487 Data Compression Techniques (Spring 2012)
Exercises 5 (22 February)

Solve the following problems before the exercise session and be prepared to present your solutions at the session.

1. A bit vector representation does not have to implement the full set of operations directly. It is possible implement some operations by calling other directly implemented operations. Show how to implement:

   (a) access using rank-1

   (b) rank-1 using select-1

   (c) select-1 using rank-1

   (d) rank-0 using rank-1

   (e) select-0 using any of the other operations

   The implementations do not need to work in constant time but should be as fast as possible.

2. The lecture notes show how to implement a sparse array using a bit vector with support for rank and select operations. Other possible implementation methods include binary trees and hashing. Compare these three methods with each other.

3. The searchable prefix sum technique in the lecture notes requires that the values are positive. Generalize the technique for non-negative values, i.e., allow zeros in the sequence.

   Hint: Use a bit vector of length $u + n$ and a full set of rank and select operations.

4. Let $\pi = [5, 6, 0, 7, 2, 8, 3, 9, 1, 4]$ be a permutation.

   (a) Find a string S with the smallest possible alphabet that represents π as described in the lecture notes.

   (b) Compute $\pi(2)$, $\pi(3)$, $\pi(9)$, $\pi^{-1}(2)$, $\pi^{-1}(3)$ and $\pi^{-1}(9)$ using the representation of part (a).

5. Generalize the succinct bit vector rank technique from the lecture notes for larger alphabets. For a sequence $S[0..n)$ over an alphabet of size $\sigma = \mathcal{O}(\log \log n)$, it should support rank in constant time using $o(n \log \sigma)$ bits in addition to the sequence S.

6. (a) Give a wavelet tree for $S = \texttt{senselessness}$.

   (b) Simulate the WT-select algorithm to compute $\text{select}_S(\texttt{s}, 3)$.