Przemysław Uznański March 13, 2018

Advanced Data Structures

Spring Semester 2018

Exercise Set 4

We aim at designing succint datastructure for storing *trits* (3-state version of bits). In particular, we want to store $X \in \{0,1,2\}^n$ using $\log_2 3 \cdot n$ bits + small overhead, having fast access (read or write) to any $X[i]$. In below, we assume (standard RAM model) that memory is composed of words of length $w = \Omega(\log n)$ bits, on which standard arithmetic operations take $\mathcal{O}(1)$ time.

Exercise 1:
Show that $\log_2 3 \cdot n$ bits are necessary to store n tris. Show that $\mathcal{O}(n)$ bits are enough to store n trits with $\mathcal{O}(1)$ time access.

Exercise 2:
Show that for some $n' = \Theta(w)$, n' trits can be stored using $\log_2 3 \cdot n' + \mathcal{O}(1)$ bits, providing $\mathcal{O}(1)$ time access.

Hint: Interpret $X \in \{0,1,2\}^{n'}$ as $X \in [0 \,..\, 3^{n'} - 1]$.

Exercise 3:
Show that n trits can be stored using $\lceil \log_2 3 \cdot n \rceil + \mathcal{O}\left(\frac{n}{w}\right)$ bits, providing $\mathcal{O}(1)$ time access.

Exercise 4:
Show that n trits can be stored using $\lceil \log_2 3 \cdot n \rceil + \mathcal{O}\left(\frac{n}{wt}\right)$ bits, providing $\mathcal{O}(t)$ time access, for any $1 \le t < n/w$.

All previous exercises are based around idea that any coding from universe X uses $\lceil \log_2 |X| \rceil$ bits, which wastes $\mathcal{O}(1)$ bits. Below we show, that for trits, the tradeoff time $\cdot$ overhead $= \mathcal{O}(n/w) = \mathcal{O}(n/\log n)$ can be improved significantly.

Definition. *For any universe X, and appriopriately chosen integeres K, M, an injection $X \to 2^{[M]} \times [K]$ is called* spill-over *representation of X. For any $x \in X$, the value of $k \in [K]$ from its representation is called its* spill*.*

Exercise 5:
Show, that for any X and $r < |X|$, there is a spill-over representation of X with $r \le K \le 2r$ such that $|X| \le K \cdot 2^M \le (1 + \frac{1}{r})|X|$. Moreover, show that encoding/decoding can be done with few arithmetic operations.

Spill-over representation wastes only $\log_2(1 + 1/r) = 1/\mathcal{O}(r)$ bits.

Exercise 6:
Show that n trits can be stored using $\lceil \log_2 3 \cdot n \rceil + \mathcal{O}\left(\frac{n}{w^2/\log w}\right) + \text{polylog}(n)$ bits, providing $\mathcal{O}(1)$ time access.

Hint: Take r to be constant to be fixed later. Use two-level scheme:

- On the first level, store $\Theta(w)$ trits in spill-over representation, for some $K \in [r \mathinner{..} 2r]$. Keep bits on this level.
- On the second level, store spills, grouped in the same way as trits in Exercises 2-4.

Count the number of bits wasted in each place.

Exercise 7:
Generalize Exercise 6 into: $\lceil \log_2 3 \cdot n \rceil + \mathcal{O}\left(\frac{n}{w^t}\right) + \text{polylog}(n)$ bits for n trits, with $\mathcal{O}(t)$ time access (for constant t).

Exercise 8:
Show that data structures from all previous exercises support writes in the same time as reads.