Computer Science 222: Succincter

Tuesday, December 11, 2012

Professor Mitzenmacher

Dan Bradley (dbradley@college), Saagar Deshpande (sdeshpande@college)

Final Project

****

1 Abstract

Using Mihai Patrascu's 2008 paper "Succincter", we implement a way to store trits (trinary values) within 1.05% of the ideal space of $n * log_2(3)$ while having lookup in $O(t)$ time, where t is the depth of our data structure. We find that this is both a fast and space efficient data structure with room for extension past simply storing trits.

2 Introduction

There are few effective methods for storing trits. In this paper, we focus on three of them: the naive method, arithmetic coding, and our succincter implementation.
The naive method is simple. As a trit has more information than a bit, but less than two bits, store each trit using two bits. Obviously this is not the most space-efficient method, you could encode $\frac{4}{3}$ as much information in the same space, so it is clearly wasteful. It is however, very fast, both on encode and decode. Encode is strictly linear, with very small constants. The encoder we used took 3.9 milliseconds to encode 500000 entries, and 13.7 milliseconds to decode 500000 entries. Arithmetic coding is much more effective at reducing size. By reducing the

3 Implementation

4 Results and Analysis

Conclusion

****

Appendix

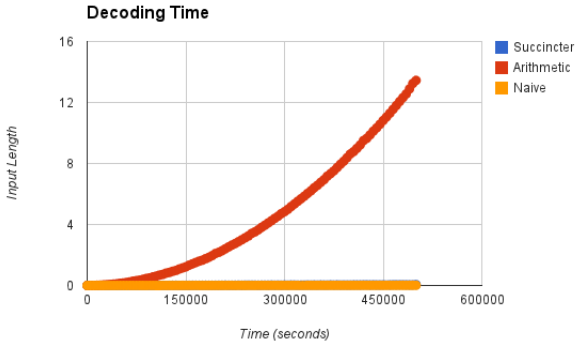

Figure 1: A figure

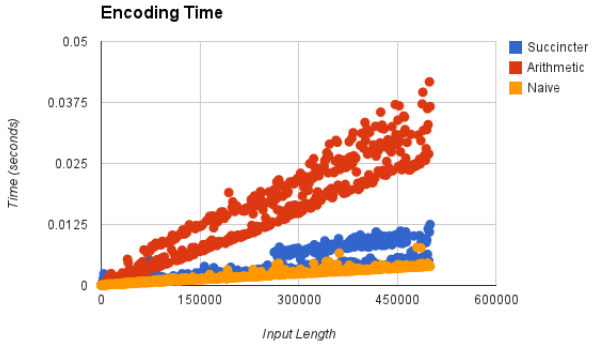

Figure 2: Another figure