582487 Data Compression Techniques (Spring 2015)
Exercises 6 (February 25th)

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

1. Write down the level-order representation of the following two binary tries.

![Binary Trees](image1.png)

2. In Exercise Set 4 you augmented the RLZ parsing of a string so that later an arbitrary character \( i \) in the original string could be accessed in \( O(\log z) \) time, where \( z \) is the number of phrases produced by the parsing. Show how to improve the character access time to \( O(1) \). How has your modification affected the space usage?

3. Back in Lecture 3 we looked at the Simple-9 word-aligned integer code. Last lecture we saw how we could get random access to arrays of integers encoded with Elias \( \gamma \) and vByte codes by first rearranging the bits in the codes appropriately, and then adding rank and select data structures.

Using the techniques you have learnt about in the last two weeks, devise a way to provide random access to arrays encoded with the Simple-9 scheme. What is the access time of your data structure? What space overhead does it add to the normal Simple-9 encoding?

4. Recall that the basic version of the Relative-10 word-aligned integer code uses four different selectors: same as last, one bigger, one smaller, and biggest. Use bitvectors to provide constant-time access to the \( i \)th selector in an array encoded with the Relative-10 scheme.

5. (Challenge) Design a data structure that provides random access to interpolative binary codes. (Hint: group code words by length).