Problem 5 Due: Thursday, Mar. 11

Be sure to read the instructions on the assignments section of the class web page.

**Cartesian trees in linear time.** Show that a Cartesian tree for an array \( A[1, \ldots, n] \) can be computed in \( O(n) \) time.

*Hint:* One way to do this is adding the elements of \( A \) according to their order in \( A \) one after another.

**Space requirements for integer data structures.** As usual, \( u \) denotes the size of the universe. We assume that \( u \) is a power of 2.

1. Show that a van Emde Boas tree needs \( O(u) \) space.

2. How many entries are stored in the hash table of an \( x \)-fast tree in the worst case after adding \( n \) elements? In the lecture we gave a brief argument for \( n \log u \). However, this estimate was rough, since we overcounted the entries in the hash table. In particular, an entry in the hash table might be a prefix of different “keys”, and we assume that every prefix is only stored once. Give a sharper bound for the number of elements stored in the hash table in terms of \( u \) and \( n \).