1 Implicit van Emde Boas Layout

The naïve implementation of the van Emde Boas layout requires storing pointers from each node to its children, but these pointers are not actually necessary. In an implicit van Emde Boas layout, only the values at the nodes are actually stored; the data structure consists entirely of a permuted array of values.

\[
\begin{array}{cccccccccccccc}
H & D & L & B & A & C & F & E & G & J & I & K & N & M & O
\end{array}
\]

Describe how to perform a binary search using the implicit van Emde Boas layout in \(O(\log_B N)\) memory operations and \(O(\lg N)\) time.

2 Cache-oblivious heapsort

Describe a cache-oblivious heap. What is the cost of inserts and lookups in your heap? What is the I/O complexity of sorting using your heap (assuming a memory of size \(M\))? How does this compare to \(M/B\)-way mergesort?

3 Sorting with buffer trees

Imagine a tree something like a \(B^*\)-tree, but nodes have size \(\Theta(M)\) and the fanout is \(\Theta(M/B)\). How long would it take to sort \(N\) items by inserting them into the tree and then enumerating the contents of the tree? How does this compare to \(M/B\)-way mergesort?

4 Hash tables in external memory

What is the I/O complexity of lookups and inserts in a hash table with linear probing?