6.851 Advanced Data Structures (Spring'12)

Prof. Erik Demaine   TAs: Tom Morgan, Justin Zhang

Problem 7   *Sample solution*

1. Construct a lookup table over the first $\lg n$ bits in a word. Each entry in the table will point to a vEB (built with perfect hashing) over the last $w - \lg n$ bits of all words whose first $\lg n$ bits match the table index. Additionally, each table entry will contain a pointer to the previous and next table entry whose vEBs are not empty.

   The procedure for a successor operation is then to first use the first $\lg n$ bits of the input word to find the relevant table entry. If the vEB for that entry contains a successor for the input word, we return it. Otherwise, use the table's pointer to the next non-empty vEB and return the smallest key in that vEB. The running time for each vEB lookup is $O(\lg(w - \lg n))$ since each vEB is over words of size $w - \lg n$. The total space used is $O(n)$ since the table is of size $O(n)$ and each key is in only one vEB.

2. We will construct a normal vEB structure (built with perfect hashing), except once the recursion gets down to a key size less of at most $\lg \lg n$, we will stop and instead use a complete lookup table. Specifically, the table will have $\lg n$ entries, one for each bit string of length $\lg \lg n$ and each entry will have the predecessor and successor for that query. The running time of queries is clearly $O(\lg \frac{w}{\lg \lg n})$ since the key length halves with each step in the recursion until it stops at $\lg \lg n$, and then an $O(1)$ lookup occurs. The space used is $O(n \lg n)$ since each lookup table has size $\lg n$ and there are at most n such tables.