6.851 Advanced Data Structures (Spring'12)

Prof. Erik Demaine

Problem 6 *Sample solution*

Concise van Emde Boas. We can shave off a factor of $\lg u$ bits of space through indirection. Divide the universe into chunks of size $\lg u$, corresponding to the last $\lg \lg u$ bits of the word. We will maintain a van Emde Boas structure over the first $\lg u - \lg \lg u$ bits. For each chunk, we maintain a single word to represent it. To insert into a chunk, simply set the corresponding bit to 1, and to delete, set it to 0. To find a successor or predecessor in a chunk, shift out the corresponding query bit and then find the least significant or most significant bit (as described in class).

Whenever we insert an element, insert its first $\lg u - \lg \lg u$ bits into the summary vEB. When we delete, if the chunk we delete from empties then we delete from the summary structure as well. To find a successor, first check the corresponding chunk for a successor, and if one exists return it, otherwise search the summary structure for the successor chunk and return the smallest element in it.

All operations run in $O(\lg \lg u)$ time since they take a constant number of operations in the vEB structure and all work in the chunks take constant time. The summary vEB takes $O(u \lg u / \lg u) = O(u)$ bits of space, and the chunks take $O(u)$ bits of space since there are $u / \lg u$ chunks, and each takes $\lg u$ bits. Thus the total structure takes $O(u)$ bits of space.

Union-Split-Find. We will maintain two van Emde Boas structures, A and B. A consists of the interval start points, and B consists of the interval end points. We perform the operations as follows:

- `make`(a, b) : Insert a into A and b into B.
- `union`(a, b, c) : Delete b from A and B.
- `split`(a, b, k) : Insert k into A and B.
- `find`(k) : Let a be the predecessor of k in A, b be the successor of a in B. If k is in the range $[a, b)$, return it, otherwise report that no interval contains k.

All operations require 2 vEB insert/delete/queries, thus they each take $O(\lg \lg u)$ time.

6.851 Advanced Data Structures

Spring 2012