CS 466/666
Brief Notes on
Reducing Space Requirement of vEB Priority Queue to O(u) Bits

The Van Emde Boas "priority queue" as initially stated uses O(u) words: there are O(u) nodes in an implicit balanced binary tree, each may have several pointers and counters, furthermore we may index into descendents of various nodes. Reducing the space to O(u), without significantly altering the run time, can be achieved as follows:
Maintain a bit vector (call it B) of length u, B[i] is1 if i is present. (We are essentially stuck with this space, especially if about half the elements are present)
The key trick is to reduce the domain of the "usual form" of the structure. For some value r, we maintain the "usual vEB structure" with element i present iff at least 1 element in the range B[ri] through B[ri +r-1] is present. So we have a vEB structure on a universe of size u/r. Call this structure V, V will require O((u/r) lg (u/r)) bits.
Let r = lg u, then the space requirement for the full structure is O((u/r) lg u)= O(u) bits for V and another u bits for B.
The issue that remains is the interplay between the two structures.

- To insert a new value i, set B[i] = 1 and insert floor(i/r) into V, if it is not already present. Deletions are similar, the greater concern if finding the first 1 to the left (or right) of B[i], if there is such an element within r of i.
- Consider the case in which we want the first bit to the right of a given spot, clearly going left is analogous. There are several approaches, one is to have a table giving the answer for every bit vector of length r, which gives the location of the leftmost 1. We then use the r bits starting in position B[i] to index into this table. With r = lg u, this requires a table of size 2^r = u, but unfortunately each entry is lglg u bits in length (as the answer can be as large as lg u, i, that is lglg u bits to represent). This is annoying, but easily fixed by having the table deal with indices of r/2 bits, so the space requirement is $2^{r/2}$ lglg u = o(u). We now ask whether the r/2 positions following B[i] are all 0's: if not look up the r/2 bits following B[i], otherwise the r/2 bits after B[i+r/2].

Setting r = lg (u /2) for V would also work and avoid the two step search in B.