

**1. When to use selection algorithms.**

Do problem 9-1.

**2. Shuffling.**

Read the part of section 5.3 on randomly permuting arrays by sorting, and do problem 5.3-6.

**3. Treaps.**

The idea used for randomly permuting by sorting in Section 5.3 can also be used to build a binary search tree. Let  $V[i]$  be the value of item  $i$ , and also give each item a randomly chosen key  $P[i]$ . Sort the items by their  $P[i]$  keys. The item  $i$  with smallest  $P[i]$  becomes the root of the tree. We assign the items  $j$  with  $V[j] < V[i]$  to the left subtree and  $V[j] \geq V[i]$  to the right subtree, and construct the subtrees recursively. Notice that along each path from the root to a leaf the  $P[i]$  keys are increasing, just like a min-heap. For this reason this data structure is called a treap. Treaps are discussed in problem 13-4.

- a) Do problem 13-4, part b) only. The analysis of randomized hiring should come in handy.
- b) Describe how to delete an item from the treap.

**4. Direct addressing.**

Do problem 11.1-4.