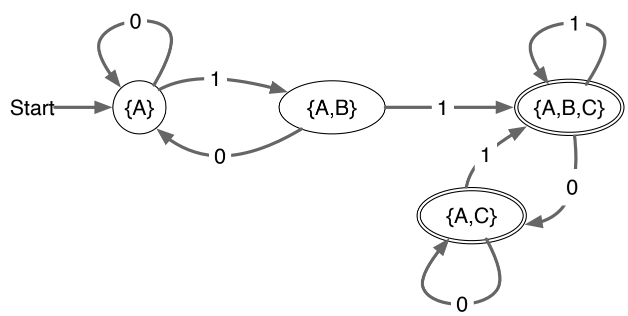
Practice Midterm Solution

Finite Automaton:

1. Since there are two transitions from State A on 1 (on to itself, on to B), the automaton is non-deterministic.
2. The states of the deterministic automaton are subsets of the set of states, but we only create the ones that we actually need.

State	on 0	on 1
{A}	{A}	{A,B}
{A,B}	{A}	{A,B,C}
{A,B,C}	{A,C}	{A,B,C}
{A,C}	{A,C}	{A,B,C}

The final / accepting states are the ones with C in them and $\{A\}$ is still the starting state.



You can surmise that an optimization would merge the two final states.

Sorting Networks:

1. The number of comparisons needed to find the maximum of *n* values is $n-1$.
2. Divide the array into groups of eight (linear work). Then determine the maxima of each group (time $\sim n/8$) and put them into an array. Then apply the algorithm recursively on the array of maxima.
3. $T(n) = T(n/8) + Cn$
4. $\log_8 1 = 0$. Therefore, we compare the linear function with $n^0 = 1$. We are in case 3 with an ϵ between 0 and 1. Since $an/8 < \frac{1}{2}n$, regularity is given. Thus, $T(n) = \Theta(n)$.

(Comment: this is no surprise, what is more interesting is to count the number of steps using the sorting network. These are $Cn + C\frac{n}{8} + C\frac{n}{8^2} + \ldots = \frac{8}{7}C$.)

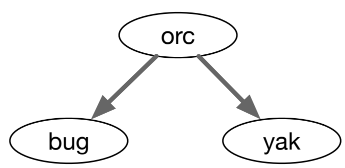
Recursive Algorithm

The recursive calls operate on arrays of size $\lceil \frac{n}{3} \rceil$ and there are five of them. The rest of the work is presumably constant. This gives $T(n) = 5T(n/3) + C$. By the master theorem, we have to calculate $\log_3(5) = 1.4649735207179269 = l$ and compare n^l with C. Since $C \in O(n^{l-\epsilon})$, we are in case 1 and have $T(n) = \Theta(n^l)$.

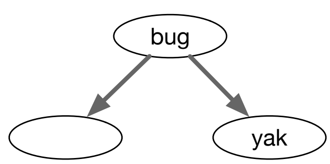
B-Tree

Before the delete operation, the B-tree needs to have three nodes and therefore at least three records. After the deletion, the B-tree records need to fit into a single node, and therefore there are at most two records. Any B-tree with three records will do.

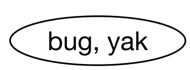
Before deletion:



If we delete any record, but let's pick 'orc', we swap 'orc' with its predecessor, and delete from there. This gives a node with underflow:



Obviously, a rotate is impossible, so we have to merge:



Linear Hashing:

A linear hash table with $10 = 2^3 + 2$ buckets has split pointer 2 and level 3. We insert the record with key hash 5 into Bucket $5 \,\%\, 8 = 5$, with key hash 6 into Bucket $6 \,\%\, 8$, with key hash 7 into Bucket $7 \,\%\, 8 = 7$, with key hash 8 into Bucket 8, with key hash 9 into Bucket 9 and with key hash 10 into Bucket $10 \,\%\, 16 = 10$.