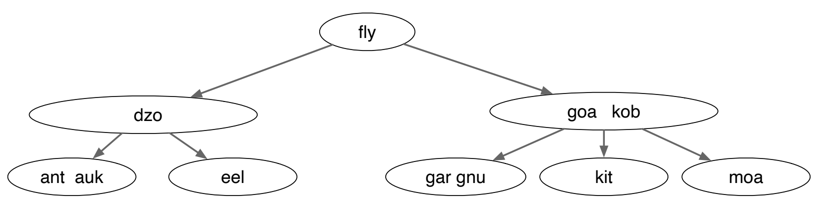
Solutions

(1) On 0, we transition to $\{C\}$, on 1, we transition to $\{C, D\}$.
(2) The smallest string is 011.
(3) The inner for-loop is taken $\lceil \frac{n}{2} \rceil$ times and runs n^2 times, for a total of $\lceil \frac{n}{2} \rceil n^2$ times, which dominates the performance. Therefore, the runtime is $\Theta(n^3)$.
(4) If $n = 4m$, we have $4m$ comparisons in the groups and an additional $2(m-1)$ comparisons among the maxima and minima for a total of $6m - 2 = 6\frac{n}{4} - 2$, which is exactly the same as for the algorithm from lecture.
If $n = 4m + 2$, we still have $4m$ comparisons in the groups of four, one comparison in the left-over group with two elements, and then $2m$ comparisons to determine the maximum of the $m + 1$ group maxima and the minimum of the $m + 1$ group minima. This gives us $4m + 1 + 2m = 6m + 1 = 6\frac{n-2}{4} + 1 = \frac{3}{2}n - 2$ comparisons, which is still the same.
(5) We recursively operate eight times over arrays of size 2/5, which gives us $T(n) = 8T(\frac{n}{5/2}) + C$ for the runtime. Since $\log_{(5/2)}(8) \approx 2.269$, $C \in O(n^{2.200})$, we are in Case 1 of the Master Theorem, and $T(n) = \Theta(n^{2.269})$.
(6) Since $20 = 2^4 + 4$, both split pointer and level are four. For the address, we first calculate $40 \,\%\, 16 = 8$ and find that 8 is larger than the split pointer. Therefore, the record goes to bucket 8.
(7) After deleting "cob", the middle node on the left side is empty. There is no possibility to rotate into it, so we have to merge with one of the neighbors. If we merge with the left neighbor, than "auk" goes down and forms a new node with "ant". Its parent only has "dzo" in it.



(8) Every time we run the loop, two marbles get removed and replaced by another marble. The total number of marbles decreases by one for each loop iteration and therefore reaches eventually one.

If we obtain m1 and m2, then there are four cases: Case 1: m1 and m2 are blue. They will be replaced by a red marble and the number of blue marbles has gone done by two. Case 2: m1 is blue and m2 is red. Then m1 gets thrown back and the number of blue marbles has not changed. Case 3: m1 is red and m2 is blue. m2 is thrown back and the number of blue marbles has not changed. Case 4: m1 and m2 are red. In this case, the number of blue marbles stays the same.

If the number of blue marbles is even, then its parity never changes. Therefore, if there is only one marble left, the number of blue marbles is still even, which can only happen if there are none. Therefore, the remaining marble has to be red.