Question 1. Transform the graph G into a new graph $\hat{G}$ that has only edge capacities, by "stretching" each vertex v of G into an edge of capacity $c(v)$ whose tail is a new vertex v^- and whose head is a new vertex v^+. An edge (v, w) of G becomes edge (v^+, w^-) in $\hat{G}$, having same capacity as (v, w) had in G. The vertices of $\hat{G}$ do not have capacities, only its edges have capacities, hence a maximum flow in it can be found using the standard algorithm for that problem. Correctness follows from the fact that any legal flow in $\hat{G}$ has a corresponding legal flow in G, and vice-versa.

Question 2.

1. First, use the linear-time pattern matching algorithm with xx as text and x as pattern, to find all occurrences of the pattern in the text. Next, choose the leftmost occurrence of the pattern in the text that starts at a text position whose index is of the form $2^k + 1$. The answer α is the prefix of x of length 2^k.

2. Check whether the left half of x equals its right half: If not then return x, otherwise (if they're equal) return the answer returned by a recursive call on the left half of x.

Question 3. Do a depth-first search of G, during which you compute for every vertex v the set NT_v of non-tree edges that are between v and an ancestor of v in the depth-first tree (i.e., those whose other endpoint has smaller depth-first number than v's); observe that NT_v is empty if v is the root or is a child of the root, so there are at most $n - 2$ non-empty NT_v sets. Next, for each non-empty NT_v set, keep only the edge (v, w) whose w has smallest depth-first number (i.e., is closest to the root) and throw away all the other edges of NT_v. The surviving edges therefore include at most $n - 2$ non-tree edges, and the $n - 1$ tree edges, for a total of at most $2n - 3$ edges (hence $c = 2$). The new graph is still biconnected because, in the biconnectivity algorithm we covered in class, the final "label" of a vertex does not depend on any of the edges we removed (if it did then we'd have a contradiction to the definition of the label function).

Reminder: The December 4 class is canceled (the December 6 class is not canceled).