(2 points) 1. We consider a generalisation of the range maximum query in which we have an arbitrary semigroup S and an array $A[1..n]$, where $A[i] \in S$ for $i = 1, 2, \ldots, n$. We would like to preprocess A for range queries, that is, computing any $A[i] \cdot A[i+1] \cdot \ldots \cdot A[j]$ (where $\cdot$ is the binary operation associated with S). Any s-space and k-query such structure can be seen as a collection of s sets $X \subseteq \{1, 2, \ldots, n\}$ for which we store the corresponding sum $\sum_{x \in X} A[x]$, so that later we can represent any interval $[i, i+1, \ldots, j]$ as a union of k disjoint sets in our collection.

   (a) Write down the details of the $\mathcal{O}(n)$-space and $\mathcal{O}(\log^* n)$-query structure that we have discussed in the lecture and observe that it can be used for an arbitrary semigroup.

   (b) For any function $f(x)$ such that $f(x) < x$ we can define another function f^* as follows:

$$f^*(x) = \begin{cases} 0 & \text{when } x \leq 1, \\ 1 + f^*(x) & \text{otherwise.} \end{cases}$$

   (so, the smallest number of times we need to apply f on x to reach 1). Then $\alpha(x)$ as follows:

$$\alpha(x) = \min\{i : \log^{\overbrace{* * \cdots *}^{i}}(x) \leq i\}.$$

   Design an $\mathcal{O}(n)$-space and $\mathcal{O}(\alpha(n))$ structure (again, for an arbitrary semigroup).

   (c) Prove that for $k = 2$ the space needs to be $\Omega(n \log n)$. Show a matching upper bound.

(difficult) (d) Show that, if the space is restricted to be linear, $\Omega(\alpha(n))$ is the best possible query time.

2. Show that any structure capable of answering range maximum queries without access to the original array (recall that the answer to a query is the position of any minimum) must take at least $2n - o(n)$ bits.

3. Design an efficient data structure for two-dimensional range maximum queries: we are given an array $A[1..n][1..m]$ that needs to be preprocessed for "what is the maximum in a rectangle $[x_1, x_2] \times [y_1, y_2]$". Focus on constant-query structures, and try to make the space close to linear (that is, $\mathcal{O}(n \cdot m)$), with one or two logarithms.