(1) 1. Show how to store a set $S \subseteq [m]$ in $B + \mathcal{O}(m \log\log m / \log m)$ bits of space, where $B = \log \binom{m}{n}$, so that membership and rank queries take constant time. Hint: split the universe into blocks of length b, where $m = b \cdot s$. Then, denoting the size of the intersection of S and the i-th block by x_i, observe (and use) that $\sum_i B_i < B + s$, where $B_i = \log \binom{b}{x_i}$.

(1) 2. Consider a balanced sequence of brackets, e.g. $()(()())$ of length n. Show how to store them in $n + o(n)$ bits of space (in a systematic structure) to implement findclose(i))/findopen(i) query, which return the corresponding closing/opening bracket.

(1+1) 3. We want to store n natural numbers $x_1, x_2, \ldots, x_n$. Show how to save them in $\sum_i 2 + 2\lfloor \log_2 x_i \rfloor$ bits, so that given an index i, we can return x_i in $\mathcal{O}(1)$ time. Try to further optimise the total number of bits used by your solution and show (some) lower bound.

(1) 4. Given an ordered collection of n items, the i-th item having weight w_i and $\sum_i w_i = W$, show how to arrange them in a BST such that the depth of the i-th item is $\mathcal{O}(1 + \log(W/w_i)$.