(1) (a) Give a high level description of a (deterministic) TM that, given input $w_1\#w_2\#\cdots\#w_\ell$, with ‘elements’ $w_i \in \{0,1\}^*$ for $1 \leq i \leq \ell$, accepts if $w_i \leq w_j$ for all $1 \leq i < j \leq \ell$, and rejects otherwise. Then give a formal description of this TM.

(b) Suppose that the TM is provided with an output tape, and at each transition it can either write a symbol to the output tape and move right, or can not write on the output tape and therefore stay in the same place. Give a high level description of TM with output tape that accepts if the input is sorted and rejects if it is not (as in the (a) part), but also writes the elements in sorted order on the output tape, again separated by $\#$.

(2) Sipser 3.9.

(3) (a) Sipser 3.12.

(b) A liberal TM is the same as a deterministic TM, but instead of moving the tape head left or right, it can keep the tape head in the same place or move left. What languages does a liberal TM accept? Explain your answer!

(4) Sipser 3.15 (c), (e) and Sipser 3.16 (b), (c).

(5) A graph $G = (V,E)$ is a finite set $V$ of vertices, and a finite set $E$ of edges, where each edge is a pair $\{x,y\} \subseteq V$. An adjacency matrix for a graph is a $|V| \times |V|$ matrix $A = (a_{ij})$ with entries from $\{0,1\}$, where $a_{ij}$ is 1 if $\{i,j\} \in E$ and is 0 otherwise. A subset $W \subseteq V$ is a clique if $a_{xy} = 1$ for every $x,y \in W$ with $x \neq y$.

(a) Describe a way to encode an adjacency matrix $A$ and a subset $W$ as a Turing machine input.

(b) Using this encoding, give a high level description of a deterministic Turing machine that decides whether $W$ is a clique in $A$.

(c) Give a formal description for your method in the (b) part.

(d) Using this checking method for cliques as a ‘subroutine’, give a high level description of a deterministic Turing machine that decides whether $A$ has a clique with at least $k$ vertices; note that $k$ is also to be provided on the input tape, and you must determine how.