1. Solve the following instance of the 0/1 knapsack problem using dynamic programming:

<table>
<thead>
<tr>
<th>Weight</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Profit</td>
<td>10</td>
<td>15</td>
<td>25</td>
<td>12</td>
</tr>
</tbody>
</table>

The capacity of the knapsack $m = 5$.

2. Let $A$ be the adjacency matrix of a directed graph $G$. The reflexive transitive closure $A^*$ is a matrix with the property $A^*(i, j) = 1$ iff $G$ has a path, containing zero or more edges, from $i$ to $j$. $A^*(i, j) = 0$ otherwise. Present an $O(M(n) \log n)$ time algorithm to compute $A^*$, where $M(n)$ is the time needed to multiply two $n \times n$ boolean matrices.

3. Present an $O(|V|)$ time algorithm to check whether a given undirected graph $G(V, E)$ is a tree. The graph $G$ is given in the form of adjacency lists.

4. Present a common CRCW PRAM algorithm that finds the maximum of $n$ arbitrary elements in $O(1)$ time using $n^{1+\epsilon}$ processors for any fixed $\epsilon > 0$.

5. The inputs are an array $A$ of $n$ elements and an element $x$. The goal is to rearrange the elements of $A$ such that all the elements of $A$ that are less than or equal to $x$ appear first (in successive cells) followed by the rest of the elements. Give an $O(\log n)$-time $\frac{n}{\log n}$-processor CREW PRAM algorithm for this problem.

6. Let $\pi_2$ be a problem for which there exists a deterministic algorithm that runs in time $2^{\sqrt{n}}$ (where $n$ is the input size). Prove or disprove:

   If $\pi_1$ is another problem such that $\pi_1$ is polynomially reducible to $\pi_2$, then $\pi_1$ can be solved in deterministic $O(2^{\sqrt{n}})$ time on any input of size $n$.

7. Assume that there is a polynomial time algorithm CLQ to solve the CLIQUE decision problem. Show how to use CLQ to determine the maximum clique size of a given graph in polynomial time.