**Design and Analysis of Algorithms**

**Question #1:**
Let $A[1...n]$ be an array of $n$ distinct numbers. If $i < j$ and $A[i] > A[j]$, then the pair $(i, j)$ is called an *inversion* of $A$.

- List the five inversions of the array $A[2, 3, 8, 6, 1]$.
- Give an algorithm that determines the number of inversions in any permutations of $n$ elements in $A$ in $O(n \log n)$ time. Explain the analysis of runtime.

**Question #2:**
Suppose you have one machine and $n$ jobs, $a_1, \ldots, a_n$. Each job $a_j$ has processing time $t_j$, profit $p_j$, and deadline $d_j$. The machine can only process one job at a time and that job must run uninterruptedly until completion. If job $a_j$ is completed by deadline $d_j$, you receive profit $p_j$, but if it is completed after, you receive nothing. Assuming all processing times are integers between 1 and $n$ and $d_j \geq t_j$ for all jobs, give an algorithm for computing the maximum profit you can make.

What is the run time of your algorithm? Justify the run time.

**Question #3:**
Suppose that a graph $G$ has a minimum spanning tree (MST) already computed. What is the runtime complexity of updating the MST if you add a new vertex and incident edges to $G$? Justify the complexity.