1. (30 points) Suppose that you are given a weighted graph $G = (N, A)$, vertices $\sigma, \tau \in N$, function $\text{length} : A \rightarrow \mathbb{R}^+$, and function $d : N \rightarrow \mathbb{R}^+$ such that for each $v \in N$, $d(v)$ is the length of a shortest path from $\sigma \rightarrow v$. Show how to construct a shortest path from $\sigma \rightarrow \tau$ in time $O(|N|+|A|)$. 
2. (35 points) Suppose you are given a directed graph $G=(N,A)$ with function $\text{length}: A \rightarrow \mathbb{R}^+$, and you seek the length of a shortest cycle in $G$. Find an algorithm with worst-case time complexity in $O(n^3)$ to solve this problem.
3. (35 points) A graph \( G = (N, A) \) is a Maia graph if it consists of two components and every vertex has degree 3 (every vertex has 3 edges incident with it). For example,

is a Maia graph. Give an algorithm to test if \( G \) is a Maia graph. Your algorithm should have worst case execution time in \( O(|N| + |A|) \).
1. To print the path in reverse order, 
   \( v = \tau \), 
   while (\( v != \sigma \)) { 
     print \( v \); 
     for each x adjacent to v  /* find v's predecessor x on shortest path */ 
       if (d(v)==d(x)+length(x,v)) 
         break;               /* break the for each loop*/ 
     v=x; 
   } 
   print \( \sigma \); 

2. Use Floyd's Algorithm, but start the computation with \( \infty \) along the main diagonal, that is, \( D[k,k]=\infty \) for \( n \geq k \geq 1 \). When the algorithm finishes, for each \( k \), \( D[k,k] \) is the length of a shortest cycle including node \( k \), and in linear time (time in \( O(n) \)), we can find the answer, which is the value of \( \min_{n \geq k \geq 1} \{ D[k,k] \} \)

3. Do a depth first search of \( G \) to check if it has two components. While each vertex is visited, check that its adjacency list has 3 vertices on it. Return true if and only if all the tests return true.