1. (10 pts) A sequence of \( n \) operations is performed on a data structure. The \( i^{th} \) operation costs \( i \) if \( i \) is an exact power of 2, and 1 otherwise.
   
   (a) Describe the accounting method of analysis.
   
   (b) Use the accounting method of analysis to determine the amortized cost per operation.

2. (10 pts) In the Union-Find algorithms, we need to know properties that relate the size of a tree rooted at a node \( x \) to the rank of the node \( x \).

   (a) Define rank of a node in this context.
   
   (b) Assuming the union by rank heuristic, prove by induction that \( \text{size}(x) \geq 2^{\text{rank}[x]} \).

3. (10 pts) Show that \((S, I_k)\) is a matroid, where \( S \) is any finite, nonempty set and \( I_k \) is the subset of all subsets of \( S \) of size at most \( k \), \( k \leq |S| \). Hint: you must show that all the properties of the matroid hold - what are they?

4. (10 pts) For Fibonacci Heaps, a method for cost analysis is based on the potential function
   \[ \Phi(H) = t(H) - 2 \cdot m(H), \]
   where \( H \) denotes the heap, \( t(H) \) denotes the number of trees in the root list of \( H \), and \( m(H) \) denotes the number of marked nodes in \( H \).
   Explain the operation of node insertion into an existing Heap, and show that its amortized cost, using this potential function, is \( O(1) \).

5. (10 pts)
   
   (a) Define binomial tree.
   
   (b) Prove by induction that a binomial tree \( B_k \) has exactly \( \binom{k}{i} \) nodes at depth \( i \) for \( i = 0, 1, 2, \ldots, k \).
6. (15 pts) Use Johnson’s algorithm to find the shortest paths between all pairs of vertices in the graph below. (8 pts) Show the value of \( h \) and \( \hat{w} \) computed by the algorithm - this needs the Bellman-Ford algorithm. (7 pts) Complete the computation - just for the path from 1 to 4, with full application of Dijkstra’s algorithm started from 1.

\[ \text{Figure 1 : A Graph for Johnson's Algorithm} \]

7. (10 pts) Given the flow network

\[ \text{Figure 2 : A Graph for Matching} \]

Run the Ford-Fulkerson Algorithm, with the Edmonds-Karp heuristics, on the flow network above to obtain a maximum flow. Show the residual network after each flow augmentation.