Problem 1. Run the shortest path algorithm on the following graph (Figure 1). Show all the steps, as well as all the \( d \) values of all nodes. Please start at root \( r \).

![Graph Image]

Figure 1: A graph with 7 nodes and 10 edges

Problem 2. Write a function in ruby that takes an array \( A \) of integers and separately prints the sum of the positive numbers, and the sum of the negative numbers. For example, if the array contains \([3, -2, 4, -1, 6, -1, 0, 3]\) the function prints 16 and -4.

Problem 3. Suppose we have a graph \( G \) and a boolean array \( \text{Visited} \) that is either true or false for each vertex. Write a function in ruby that takes a graph specification as an array, and prints out all the vertices that have \( \text{Visited}[i] \) set to false.

Problem 4. Trace the following code and write out the calls that are made, and the evolution of each variable. For each function call, state the value of variable \( x \), the functions it calls, and its return value.

```ruby
def f(x)
    return (x == 0 or g(x - 1))
end

def g(x)
    return (x != 0 and f(x - 1))
end
```

You should create a table of the following form, and show the results for \( f(6) \), and \( g(6) \).

<table>
<thead>
<tr>
<th>Function call</th>
<th>x</th>
<th>Calls</th>
<th>Returns</th>
</tr>
</thead>
<tbody>
<tr>
<td>( f(6) )</td>
<td>6</td>
<td>( g(5) )</td>
<td>...</td>
</tr>
</tbody>
</table>
Problem 5. Trace the following code and write out the calls that are made, and the evolution of each variable. For each function call, state the value of variables bottom, top, mid, the functions it calls, and its return value.

\[ A = [1, 2, 3, 5, 8, 13, 21, 34, 55] \]

def findIndex(s, bottom, top):
    if (bottom > top):
        return nil
    else:
        mid = (top + bottom) / 2;
        if s == A[mid]
            return mid
        elif s > A[mid]
            findIndex(s, mid + 1, top)
        else
            findIndex(s, bottom, mid - 1)
end

Function call | bottom | top | mid | Calls | Returns
findIndex(1, 0, A.length) | 0 | 9 | 4 | findIndex(1, 0, 3) | ...

Trace the code for calls $\text{findIndex}(1, 0, A.length)$, $\text{findIndex}(34, 0, A.length)$, and $\text{findIndex}(27, 0, A.length)$.