Reputation Score: In class (Handout #7, slides 12-19), we described a method to find reputation score of Internet web pages based on the links connecting them. That scheme is the basis of an algorithm called PageRank, which is used by Google for ranking Internet web pages. In this assignment, we will study an alternative way of assigning reputation scores to different web pages.

A new search engine called Elgoog-199 wants to give higher scores to the pages that are more likely to be visited by users. The idea is to simulate the way users browse web pages, and assign higher scores to the pages that are visited more frequently.

The simulation of web browsing is performed as follows. A given user starts at a randomly selected web page. Then, the user randomly clicks on one of the links leaving that page which moves the user to the next page. For simplicity we assume that all outgoing links are equally likely to be selected by the user. The user stops if there are no outgoing links.

For the rest of this assignment, you need a coin, and a die.

1) Sample Reputations [3 points]: Run the algorithm described below 20 times on the following graph, and record the number of times you visit each node of the graph. In each iteration of the algorithm, start at step A and continue till you reach step F. Then, start another iteration from the beginning (i.e. step A). Repeat this process for 20 iterations.

A. Select a node at random. To do this, simply roll the die and the corresponding node is where the user starts browsing.
B. Mark the current node as being visited (count the number of visits).
C. If the node has no outgoing links, this iteration is complete. Go to F.
D. If there is just one outgoing link, go to the next node. Continue to step B.
E. If the node has more than one outgoing link, flip a coin. If the coin comes up heads follow the link marked with “H”. If it comes up tails follow the link marked with “T”. Go to step B.
F. End of iteration.

![Sample Graph](image_url)

Figure 1 - A sample graph of web pages and the links between them.

Once you have completed this process 20 times, record the number of times you have visited each node in the following table.
<table>
<thead>
<tr>
<th>Node #</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of Visits</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

2) **Impact Graph [1 point]**: Which nodes have impact on the score calculated for node 2? What about node 4? [Hint: if a node removes one of its links, and the score of another node changes, then the former has impact on the latter.]

3) **Reputation Criteria [2 points]**: Answer the following questions based on the numbers you calculated in problem 1 of this assignment. The goal is to gain a better understanding of how the algorithm works.
   a) How do incoming links affect the score of a node?
   b) How do outgoing links affect the score of a node?
   c) How do the scores of the nodes that link to a given node affect that node’s score? For example, in our graph 1 and 2 point to 5. What is the impact of the score you recorded for 1 and 2 on the score of node 5?

4) **Going to Infinity [2 points]**: If we repeat this algorithm many times, how do you think the relative scores of nodes will be? To find get a better understanding of this, answer the following questions:
   a) Which node will have the highest score?
   b) Which node will have the lowest score?
   c) Between nodes 4 and 5, which one do you expect to have a higher score? Why?
   d) Between nodes 3 and 6, which one do you expect to have a higher score? Why?
   e) Can you give a full ranking of nodes according to the scores you predict for them?

5) **Robustness [1 point]**: One important aspect of a reputation score is its robustness to attacks. Assume that someone has a webpage and wants to get a high score in Elgoog-199’s search engine. Can he/she increase the score of his/her page simply by increasing the number of links it has to other pages? Explain.

6) **Comparison [1 point]**: Compare this algorithm with the one described in class. List some of their similarities and differences (if any). [Hint: Use questions in problem 3 for comparison.]