

Introduction to Web Science

Assignment 7

PD Dr. Matthias Thimm

thimm@uni-koblenz.de

Ipek Baris Schlicht

ibaris@uni-koblenz.de

Kenneth Skiba

kennethskiba@uni-koblenz.de

Institute of Web Science and Technologies

Department of Computer Science

University of Koblenz-Landau

Submission until: 19.01.2021, CEST 23:59

Team: Bravo

Members:

Gaurav Kumar (220200656)

Pavithree Shetty (220200661)

Nisha Sharma (220202359)

1 Generative Models for the Web (25 points)

We provide you a file called `sample_simple_english_wiki.txt`. The file contains a text snippet from Simple English Wikipedia. Your tasks are as follows:

1. Create a probabilistic generative model of text by sampling from the distribution of 1) word lengths 2) frequency of each character in the given text, similar to the one presented in the video slides¹.

Modelling choices:

- Your model should generate one word at a time according to the distribution of the word lengths.
- Within each word, generate characters according to the distribution of character frequencies.
- Your model should only consider lowercase letters [a-z] and numbers [0-9]. All uppercase letters should be converted to lowercase. Other characters such as punctuation should be excluded.

Generate a text of 5,000 words with your model. Please upload your generated text as an individual file, but do not include it in the PDF document.

2. Plot the probability distribution of word lengths of 1) the original text 2) the text you generated in one plot. Save the plot as png file.
3. Discuss the resulted plots and generated text (max half page).

For this task, you are allowed to use only `string`, `regex`, `numpy`, `matplotlib` as library.

1.1 Solution:

Generated graph and Generated text

1. Since there are no test conducted on the generative model, visually it shows that the length of words generated through the model is comparatively smaller than the actual text provided.
2. The number of generated words are 5 times more than the provided words which makes the comparison between the two text a little difficult.
3. The probability of word length for generated words are roughly 5% less than the provided words in the original text.
4. The generated words are following exponential pattern with larger length word's frequency being negligible.

¹https://en.wikiversity.org/wiki/Web_Science/Part2:_Emerging_Web_Properties/Generative_Models_for_the_Web/Sampling_from_a_probability_distribution

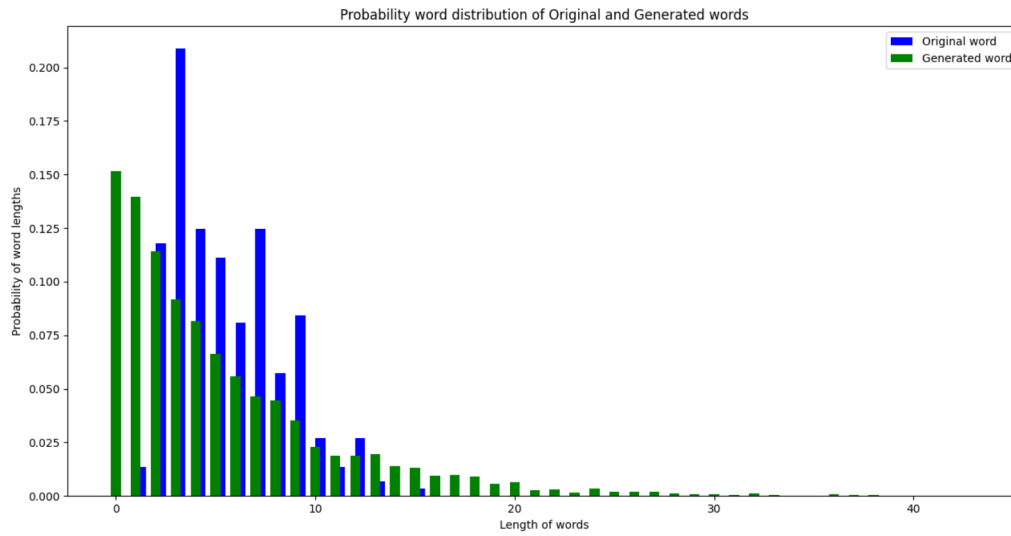


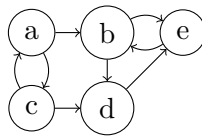
Figure 1: Probability word distribution

5. Visually comparing the probability shows that the average probability of the original words are higher than the generated word
6. This comparison can be further quantified or falsified using various tests.
7. Generated text is fairly good given that the model is quite simple and have various drawbacks.
8. The model is very simple and the generated text is not that accurate and might lack precision.

2 Directed Graphs

(30 points)

Consider the following directed graph G :



1. Write down the adjacency matrix A of the graph G . adf

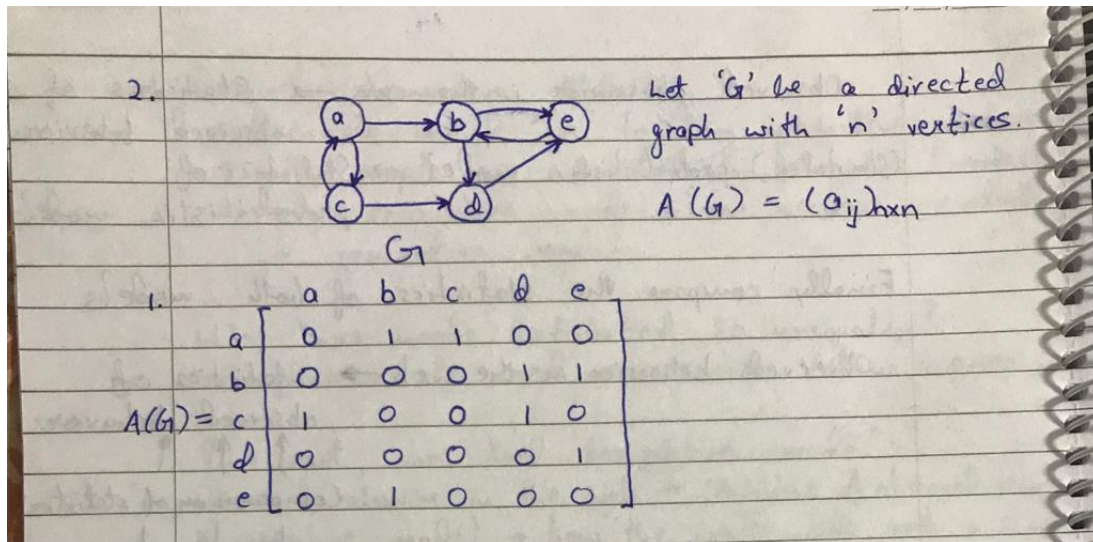


Figure 2: Adjacency matrix

2. Calculate the In- and Outdegree of every vertex.

2.	Vertex	Indegree	Outdegree
	a	1	1+1 = 2
	b	1+1 = 2	1+1 = 2
	c	1	1+1 = 2
	d	1+1 = 2	1
	e	1+1 = 2	1

Figure 3: In- and Outdegree

3. Calculate the In- and Outdegree of every vertex using counting matrix.

2. Calculate the In- and Out degree of every vertex.
 Multiply by counting matrix to A.

In-degree:

$$\begin{pmatrix} 1 & 1 & 1 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} = (1, 2, 1, 2, 1) = 1\vec{e}_1^t, 2\vec{e}_2^t, 1\vec{e}_3^t, 2\vec{e}_4^t, 1\vec{e}_5^t$$

Out degree:

$$\begin{pmatrix} 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 \\ 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 \\ 0 & 1 & 0 & 0 & 0 \end{pmatrix} \begin{pmatrix} 1 \\ 1 \\ 1 \\ 1 \\ 1 \end{pmatrix} = (2, 2, 2, 1, 1) = 2\vec{e}_1, 2\vec{e}_2, 2\vec{e}_3, 1\vec{e}_4, 1\vec{e}_5$$

Figure 4: In- and Outdegree Counting matrix

4. Highlight all strongly connected components.

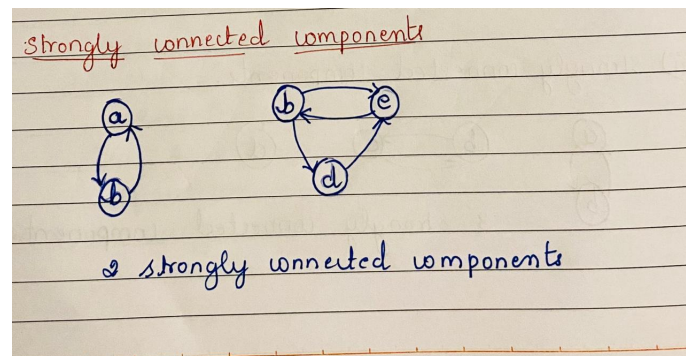
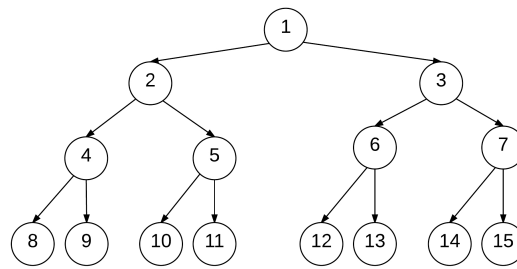


Figure 5: Strongly connected components

5. Construct **one** graph G' , which is *isomorphic* to $G.H$.

Consider the following graph H .



5. Assuming vertex 11 is our goal vertex. In which order will the nodes be expanded when breadth-first search, and depth-first search is used?

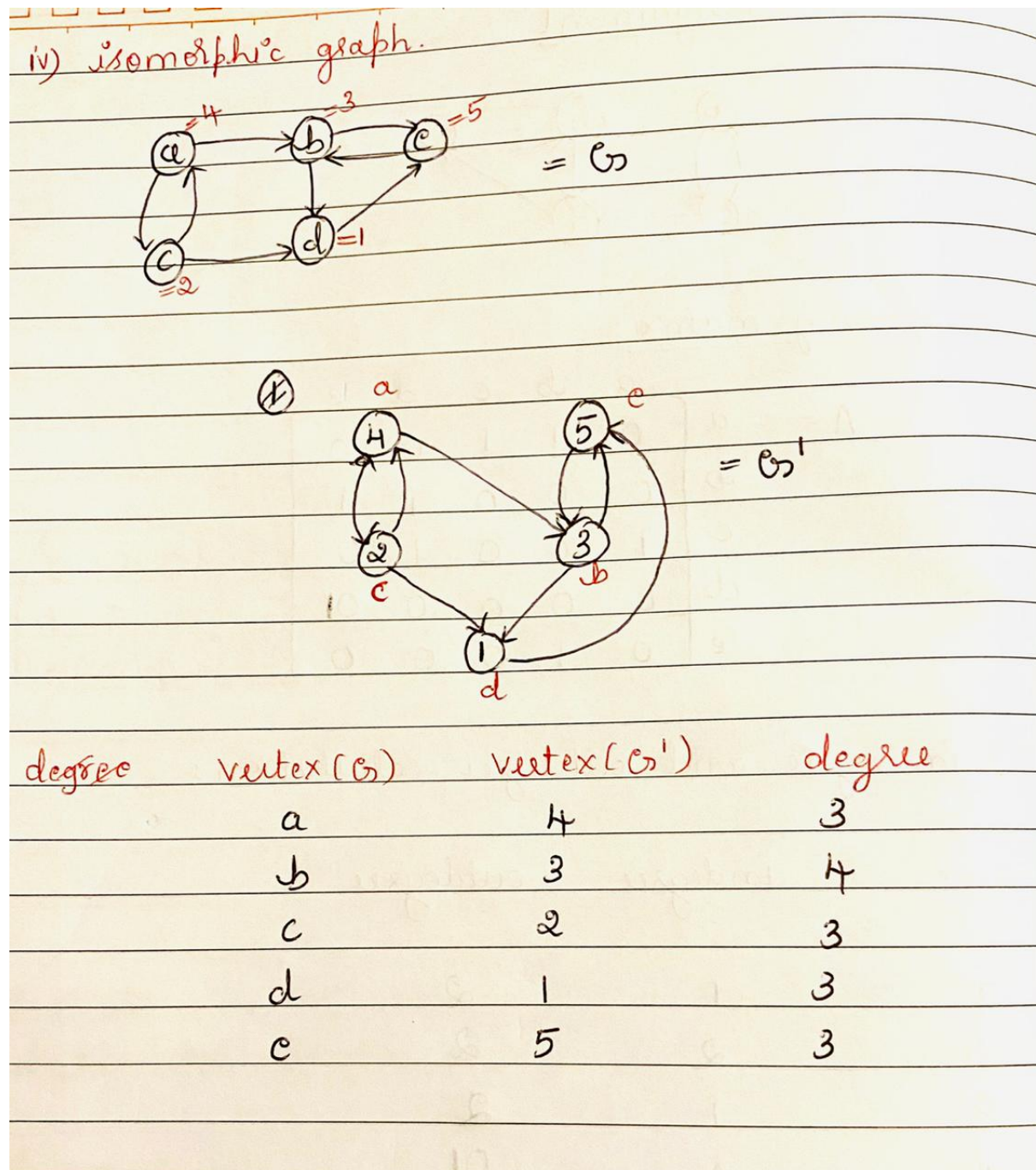


Figure 6: Isomorphic

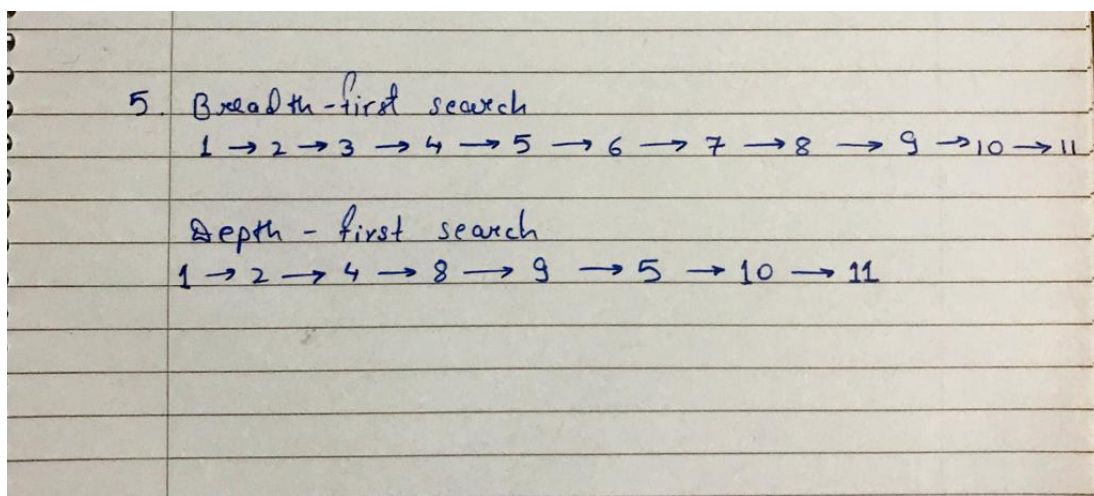


Figure 7: Breadth-first search, and Depth-first search

3 Undirected Graphs

(25 points)

Let $G = (V, E)$ be an *undirected graph*. Using Definitions 1,2,3 prove that Theorem 1 holds.

Definition 1 The function $e(v)$ of a vertex $v \in V$ returns the longest distance between v any other vertex of G :

$$e(v) = \max_{w \in V} d(v, w)$$

Definition 2 The diameter $diam(G)$ of G is the greatest $e(v)$ of any vertex in G :

$$diam(G) = \max_{v \in V} e(v)$$

Definition 3 The radius $rad(G)$ of G is the smallest $e(v)$ of any vertex in G :

$$rad(G) = \min_{v \in V} e(v)$$

Theorem 1 $rad(G) \leq diam(G) \leq 2 * rad(G)$

3) undirected graphs

Theorem 1: $\text{rad}(G) \leq \text{diam}(G) \leq 2 * \text{rad}(G)$

we have,

$$\text{rad}(G) = \min_{v \in V} e(v) \quad \text{--- def(3)}$$

$$\text{diam}(G) = \max_{v \in V} e(v) \quad \text{--- def(2)}$$

where

$$e(v) = \max_{w \in V} d(v, w) \quad \text{--- def(1)}$$

$\therefore \text{rad}(G) \leq \text{diam}(G) \quad \text{--- (1)}$

Let c be a vertex at a minimal distance from all the vertices.

$$d(c, v) \leq \text{rad}(G)$$

$$d(c, u) \leq \text{rad}(G)$$

where $u, v \in V(G)$ and d is the distance using triangular inequality,

$$d(u, v) \leq 2 \text{rad}(G)$$

$$\text{diam}(G) \leq 2 \text{rad}(G) \quad \text{--- (2)}$$

with (1) and (2) we can prove that

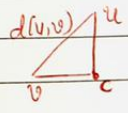
$$\text{rad}(G) \leq \text{diam}(G) \leq 2 \text{rad}(G)$$


Figure 8: Undirected Graphs