



Grundlagen der Multimediatechnik

Wintersemester 2021/22

Übungsblatt 3

15. November 2021

Wichtig: Bitte kennzeichnen Sie Ihre Abgabe mit Ihrem Namen und dem Namen Ihres Übungspartners.

Laden Sie bitte Ihre Abgabe als PDF-/ZIP-Datei auf der ILIAS-Plattform hoch. Andere Dateiformate sowie Scans von handgeschriebenen Abgaben werden nicht gewertet. Achten Sie darauf, nur kompilierbaren, kommentierten Code abzugeben. Nicht-kompilierbarer Code wird mit **0 Punkten** bewertet!

Alle Abgaben müssen folgender Namenskonvention entsprechen: `gmt_uebungXX_nachname1_nachname2.format`

Aufgabe 1: Lempel-Ziv-Welch-Kodierung (LZW)

[6 Punkte]

1. Wenden Sie die Lempel-Ziv-Welch-Kodierung, wie sie in der Vorlesung vorgestellt wurde, auf folgende Wörter an:

bonobonbon

kirschsaft

Codieren Sie die Buchstaben $a - z$ mit den Zahlen $0 - 25$ und neue Wörterbucheinträge mit ganzen Zahlen aufsteigend, beginnend mit der 26. Sie können annehmen, dass keine Zahlen in den Eingabetexten vorkommen.

[5 Punkte]

2. Welche Voraussetzung muss ein Text/Wort erfüllen, damit die LZW-Kodierung effektiv genutzt werden kann?

[1 Punkt]

Aufgabe 2: Lauflängenkodierung

[8 Punkte]

Verwenden Sie das mitgelieferte Notebook `Uebung03.ipynb` zur Bearbeitung dieser Aufgabe.

Anmerkung: Für die Vervollständigung des Skriptes sind keine externen Bibliotheken notwendig und sind somit für die Bearbeitung dieser Aufgabe unzulässig (**keine imports!**).

1. Erklären Sie, was man unter Entropiekodierung versteht und nennen Sie zwei Beispiele. [2 Punkte]
2. Erklären Sie, wie die in der Vorlesung besprochene Lauflängenkodierung funktioniert und geben Sie ein Beispiel, bei dem die Lauflängenkodierung Anwendung findet. [2 Punkte]
3. Vervollständigen Sie die `rle(word)`-Funktion. Die Funktion soll einen `string` erhalten und diesen mittels Lauflängenkodierung kodieren und zurückgeben. [4 Punkte]

Aufgabe 3: Burrows-Wheeler-Transformation

[11 Punkte]

Ziel dieser Aufgabe ist es, die aus der Vorlesung bekannte Burrows-Wheeler-Transformation zu implementieren. Verwenden Sie dafür das mitgelieferte Notebook `Uebung03.ipynb`.

Anmerkung: Für die Vervollständigung des Skriptes sind keine externen Bibliotheken notwendig und sind somit für die Bearbeitung dieser Aufgabe unzulässig (**keine** imports!).

1. Im ersten Schritt müssen Sie die `create_permutation_matrix(word)`-Funktion vervollständigen. Diese bekommt einen String als Input und soll die zugehörige Permutationsmatrix erstellen und lexikographisch sortieren. Anschließend soll die sortierte Matrix zurückgegeben werden. [3 Punkte]
2. Vervollständigen Sie anschließend die `bwt_transform(matrix, word)`-Funktion. Diese bekommt als Parameter die sortierte Matrix aus Schritt 1 sowie das ursprüngliche Wort zur Ermittlung des zugehörigen Index. Am Ende sollen der ermittelte Index sowie die letzte Spalte der Matrix als String zurückgegeben werden. [3 Punkte]
3. Implementieren Sie anschließend noch die Dekodierung in der `bwt_retransform(encoded_word, idx)` Funktion. `encoded_word` ist dabei der zuvor kodierte String und `idx` der zugehörige ermittelte Index. [4 Punkte]
4. Bei der Burrows-Wheeler-Transformation findet keine Kompression statt. Welcher Vorteil ergibt sich dennoch durch Anwendung der Burrows-Wheeler-Transformation? [1 Punkt]

Abgabe: Dienstag, 23. November 2021, 08:00 Uhr im ILIAS-System