Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - Email

Pen and Paper Task 1

Subtask A

- sun → 1, 2, 3, 4, 7
- nice → 1, 5, 6
- water → 5, 6, 8, 9
- is → 6
- beer → 10

Subtask B

- sun → 1, 2, 3, 4, 7
- nice → 1, 5, 6
- water → 5, 6, 8, 9
- is → 6
- beer → 10

Example query: *nice AND is*

1. Comparisons without skip pointers: 1 & 6, 5 & 6, 6 & 6 ⇒ 3 comparisons
2. Comparisons with skip pointers: 1 & 6, 6 & 6 ⇒ 2 comparisons

Without skip pointers we must compare the terms step by step, although 5 in **nice** is still smaller than 6 in **is**. With skip pointers we can skip the 5 in **nice** and directly go to the 6 in **nice**.

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - Email

Pen and Paper Task 2

```
    tokenize(text: string):
        token = ''
        list_of_tokens = []

        for char in text:
            if char == ' ':  #whitespace
                list_of_tokens.add(token)
                token = '' #empty string
            if char is a symbol:
                list_of_tokens.add(token)
                list_of_token.add(char)
                token = '' #empty string
            else:
                token += char

        return list_of_tokens
```

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - Email

Pen and Paper Task 3

Query: *Gates /2 Microsoft*

(Gates, 4): [{1:[3], 2:[6], 3:[2, 17], 4:[1]}]
(Microsoft, 4): [{1:[1], 2:[1, 21], 3:[3], 5:[16, 22, 51]}]

cross product = { 1:[(3,1)], 2:[(6,1), (6,21)], 3:[(2,3), (17,3)], 4:[(1,16), (1,22), (1,51)] }

From all tuples in the cross product, the tuples (3,1) and (2,3) fulfill the query's condition abs(tuple[1]-tuple[0]) ≤ 2. So the answer is: document 1, document 3.

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - Email

Programming Task 1

```
import csv, re, nltk

def index(filename: str):
    index = {}
    dictionary = {}
    postings_lists = []

    tokenizer = nltk.RegexpTokenizer(r"\w+")

    with open(filename, 'r') as file:
        reader = csv.reader(file, delimiter = '\t')
        postings = []

        #iterate through each row of the table
        for row in reader:
            (doc_id, url, pub_date, title, news_text) = row

            #tokenize and normalize news text
            #this procedure will remove symbols like !?() etc.
            #the set data structure will remove all duplicates
            news_text_norm = set(tokenizer.tokenize(news_text.lower()))

            #generate postings
            #iterate through each term
            for term in news_text_norm:
                postings.append((term, doc_id))

        #sort postings
        postings = sorted(postings[1:], key = lambda tup: tup[0])


        post_id = 0
        post_size = 0
        #iterate through postings
        for posting in postings:
            term, doc_id = posting

            if term not in dictionary:
                #upate the dictionary with the new term
                #initialize the postings size
                #save the postings id,
                #witch is the position of the postings list
                #into the postings lists
                dictionary.update({term: [post_size+1, post_id]})

                #initialize a new postings list
                postings_lists.append([doc_id])

                #update postings id
                post_id +=1
            else:
                #update size of posting
                dictionary[term][0] += 1
```

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - Email

```
54
55                  #update postings list
56                  postings_lists[-1].append(doc_id)
57
58      return dictionary, postings_lists
59
60  class Search:
61
62      def __init__(self, filename: str, index: tuple):
63          self.filename = filename
64          self.index = index
65
66      def getPostingList(self, postID):
67          postings_lists = self.index[1]
68
69          #set the index of the first postings list
70          idx = 0
71          #iterate through postings lists
72          for postings_list in postings_lists:
73              if postID == idx:
74                  return postings_list
75                  break
76              else:
77                  #update index
78                  idx += 1
79
80      def query(self, term: str):
81          dictionary, postings_lists = self.index
82          postID = dictionary[term][1]
83          postings_list = self.getPostingList(postID)
84
85          #retrive text
86          with open(filename, 'r') as file:
87              reader = csv.reader(file, delimiter = '\t')
88              postings = []
89
90              #iterate through each row of the table
91              for row in reader:
92                  (docID, url, pub_date, title, news_text) = row
93                  if docID in postings_list:
94                      return(docID, news_text)
95
96          pass
97
98      def query(self, term1: str, term2: str = ''):
99          dictionary, postings_lists = self.index
100         out_list = []
101
102         #CASE 1: only one term
103         if term2 == '':
104             postID = dictionary[term1][1]
105             postings_list = self.getPostingList(postID)
106
107             #retrive text
108             with open(filename, 'r') as file:
109                 reader = csv.reader(file, delimiter = '\t')
```

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - Email

```
                postings = []

                #iterate through each row of the table
                for row in reader:
                    (docID, url, pub_date, title, news_text) = row
                    if docID in postings_list:
                        out_list.append((docID, news_text))
        #CASE 2: two terms
        else:

            intersection_list = []

            term1_postID = dictionary[term1][1]
            term2_postID = dictionary[term2][1]

            term1_postings_list = self.getPostingList(term1_postID)
            term2_postings_list = self.getPostingList(term2_postID)

            #intersection algorithm
            for term1_docID in term1_postings_list:
                for term2_docID  in term2_postings_list:
                    if term1_docID  == term2_docID :
                        intersection_list.append(term1_docID )

            #retrive text
            with open(filename, 'r') as file:
                reader = csv.reader(file, delimiter = '\t')
                postings = []

                #iterate through each row of the table
                for row in reader:
                    (docID, url, pub_date, title, news_text) = row
                    if docID in intersection_list:
                        out_list.append((docID, news_text))
        return out_list


if __name__ == "__main__":
    filename = 'assignment1/code/postillon.csv'
    index = index(filename=filename)
    search = Search(filename=filename, index=index)

    #queries
    print('weiß AND maß')
    for item in search.query('weiß', 'maß'):
        print(item)

    print('weiß AND masse')
    for item in search.query('weiß', 'masse'):
        print(item, '\n')

    print('weiss AND maße')
    for item in search.query('weiss', 'maße'):
        print(item, '\n')

    print('weiss AND masse')
```

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - Email

```
166     for item in search.query('weiss', 'masse'):
167         print(item, '\n')
```

code/script.py

The result shows the docID with the news text. With the use of an iterator -in Python simply a for loop- I iterate through the first and second postings list to find if the two terms are found in both documents.
The differences between the two queries is that in the first one we don't need an intersect algorithm. In this case we can use the postings list of the given term directly. In the second one with two terms we have to calculate the intersection of the two postings lists and the return the text of the documents that have bouth terms in it.

Terminal OUTPUT:

```
weiß AND maß
('691', '   +++ Würde am liebsten im Erdboden versinken: Maulwurf hat das GraACben
    verlernt +++ +++ Aus Fluss: Schwimmerin weiß, woher sie Entzündungskrankeit hat
    +++ +++ Es ist ein Mädchen!: Fliege bekommt Nachwuchs +++ +++ Nur an der Nase
    herumgeführt: Masochist unzufrieden mit Domina-Dienstleistung +++ +++ Maß Capone
    : Schneider von Gangsterboss erhält Frischkäse zum Dank +++ +++ Hat Eindruck
    hinterlassen: Jürgen W. Möllemann +++ +++ Stehen auf der Gersteliste: Gerd Reide
     und Johann Hafer zu Roggen-Roll-Party eingeladen +++   Jetzt bestellen! Die
    besten Newsticker als Buch (nur 9,99ÄC): Der Postillon: +++ Newsticker +++ ')

weiß AND masse
('864', '   Berlin (dpo) - Die Polizei spricht von einem gefährlichen neuen Drogen-
    Trend: Immer mehr Berliner begeben sich seit gestern mit Bananenresten in den
    Atemwegen in ärztliche Behandlung. Intensive Recherchen des Postillon im
    Berliner Drogenmilieu haben ergeben, dass der Ursprung der mysteriösen Fälle
    offenbar bei einer 140-Kilo-Lieferung eines neuen Mode-Kokains aus Kolumbien
    liegt, die seit Dienstagvormittag im Umlauf ist. Wie der Stoff in die Szene
    geraten ist, bleibt unklar.  Nach Angaben von Dealer Henrik K.* (*Name von der
    Redaktion geändert), der mehrere Kilogramm des neuartigen Kokains kaufte, hat
    die Lieferung gravierende Mängel. "Nicht einmal gerade Lines lassen sich mit dem
     Zeug legen", klagt der 32-Jährige.  Ist unzufrieden mit dem neuen Stoff:
    Henrik K.  Einige seiner Kunden, die spekulierten, dass es sich bei der neuen,
    gelb-gebogenen Darreichungsform um Verpackung handeln könnte und nach
    stundenlangem Kampf zum Inneren vordrangen, berichten von einer matschigen Masse
    , die zwar einigermaßen weiß, aber kein bisschen feinkörnig sei und nur die Nase
     verstopfe. Lediglich ein Abnehmer, der das neue Kokain im LSD-Rausch aß, habe
    von einem wohligen Völlegefühl berichtet und wolle unbedingt mehr von dem "
    Wunderzeug" kaufen. Henrik K. schöpft daraus jedoch wenig Trost: Seit gestern
    trocknet er 3 Kilogramm der Masse in seinem Backofen, um sie anschließend zu
    zermahlen und doch noch irgendwie nutzbar zu machen. "Wenn das nicht klappt,
    kann ich nur noch schauen, ob es vielleicht intravenös funktioniert. Sonst weiß
    ich auch nicht weiter." Zwar scheint eine Überdosis mit dem neuen Stoff nicht mö
    glich, acht Konsumenten werden aber wohl bleibende Schäden davontragen. Sie
    leierten ihre Nasenlöcher beim Versuch, das Kokain ungeschält zu schnupfen,
    hoffnungslos aus.  ')

('2197', '   Osnabrück (dpo) - Müssen Fans bald auf eine liebgewonnene Tradition
    verzichten? Nachdem Schiedsrichter Martin Petersen beim Pokalspiel zwischen dem
    VfL Osnabrück und RB Leipzig von einem Feuerzeug getroffen wurde, erwägt der DFB
```

Alberto Saponaro - saponaroalberto97@gmail.com
Walter Väth - walter.vaeth@gmail.com
Chong Shen - st143575@stud.uni-stuttgart.de
Xin Pang - Email

```
    nun ernsthaft, das Werfen von Feuerzeugen in Fußballstadien unter Verbot zu
    stellen. Man befürchtet, dass von den Projektilen Verletzungsgefahr ausgeht. "
    Fußball ist ein Erlebnis für die ganze Familie. Deshalb spielen wir mit dem
    Gedanken, Feuerzeugwürfe – so schön sie auch anzuschauen sind – zu untersagen",
    erklärte DFB-Präsident Wolfgang Niersbach. "Derzeit werben wir für die
    Zustimmung der Vereine zu dieser sicherlich umstrittenen Maßnahme."  Darf bald
    nur noch zum Entzünden von Pyrotechnik verwendet werden: Feuerzeug  Fan-
    Initiativen zeigen sich empört: "Fußball lebt von Emotionen", erklärt etwa RenÄC
    Finkel vom Osnabrück-Fanclub Lila-Weiß Westerkappeln. "Da gehört das Werfen von
    Feuerzeugen aus einer anonymen Masse einfach dazu. Was kommt als Nächstes? Ein
    Verbot, den Platz zu stürmen?" Tatsächlich hat der Feuerzeugwurf Tradition. Er
    geht zurück bis Mitte des 19. Jahrhunderts, als es üblich war, Feuersteine und
    Zunderdosen auf das Spielfeld zu werfen. Ziel der Würfe war es schon damals,
    gegnerische Spieler oder den Schiedsrichter möglichst schwer zu verletzen und
    einen Spielabbruch herbeizuführen. Während der Weimarer Republik geriet der
    Brauch nahezu in Vergessenheit. Zu wenig Wucht entwickelten die damals üblichen
    Streichhölzer. Erst mit der massenhaften Einführung des Feuerzeugs erlebte der Z
    ündmittelwurf eine Renaissance. Sollte das Feuerzeugwurfverbot tatsächlich
    kommen, müssten zahlreiche Fanclubs und Ultra-Gruppierungen ihre Feuerzeugwurf-
    Bataillone auflösen oder auf andere Wurfgegenstände umlernen lassen. Auch die
    beliebten Wurffeuerzeug-Stände in den Stadien würden dann der Vergangenheit
    angehören.  ')
8
9 ('4933', '  Köln, Stuttgart, München (dpo) – Ein meteorologisches Rätsel sorgt
    derzeit in weiten Teilen Deutschlands für große Verunsicherung. Der Grund: In
    vielen Regionen fiel im Laufe des Tages eine bislang noch unbekannte nasskalte,
    helle Masse vom Himmel. Vielerorts blieb die Substanz liegen und färbte die
    Landschaft leuchtend weiß ein. Laut Behörden besteht keine Gefahr für die Bevö
    lkerung.  "Es war unheimlich", berichtet Hausfrau Renate P. aus Stuttgart dem
    Postillon. "Plötzlich hab ich was Kaltes auf der Nase gespürt. Ich schau nach
    oben und seh, dass da so weißes Zeug runterrieselt. Ganz leise. Sah aus wie bei
    mir daheim in der Tiefkühltruhe." Frau P. flüchtete sich daraufhin gemeinsam mit
     anderen Passanten in einen Laden, wo sie mehrere Stunden lang ausharrte.  "
    Maaammmiiiiii! Ich hab Angst!"  Inzwischen haben Meteorologen eine Erklärung für
     das Phänomen parat. Demnach soll es sich nicht wie zunächst vermutet um
    Vulkanasche oder radioaktiven Fallout handeln. "Vielmehr haben wir es bei der
    weißen Substanz mit einer speziellen Form von Niederschlag zu tun, wie man sie
    sonst aus dem skandinavischen Raum oder aus alten Erzählungen kennt", erklärt
    Susanne Brauer vom Deutschen Wetterdienst. "Früher hat diese Substanz in der
    Winterzeit oft weite Teile Europas bedeckt." Zwar halten die Behörden den
    Niederschlag bislang für ungiftig, doch zumindest auf Autofahrer scheint die
    unbekannte Masse einen bizarren Effekt zu haben. Viele von ihnen reagieren
    panisch und verwirrt und scheinen innerhalb kürzester Zeit selbst die
    grundlegendsten Regeln des Straßenverkehrs vollständig zu vergessen. Vorerst
    sollen deshalb die kommunalen Winterdienste sämtliche Straßen mit extrem langsam
     fahrenden Räum- und Streufahrzeugen blockieren, bis sich die Lage wieder
    beruhigt hat.  ')
10
11 weiss AND maße
12 weiss AND masse
```