ניהול נתונים באינטרנט – תרגיל מס' 2 –
Crawling, Information Extraction, Ontologies & PageRank

הוראות:
יש לעלות את הפתרונות בקובץ ZIP לMOODLE, שכולל קובץ PDF בשם answers.pdf ובו הפתרון וקבצי קוד נוספים (HTML, XML או Python) לפי הדרישה של כל סעיף וסעיף. ההגשה היא בזוגות, ורק אחד מבני הזוג יגיש את התרגיל, אך יש להקפיד לכתוב את השמות והת.ז. של שני בני הזוג בתוך הקובץ.
שם של הקובץ ZIP צריך לכלול את הת.ז. של אחד מהמגישים (למשל: HW1_123.zip).
תאריך הגשה: **17.05.2020**

שאלה 1: (Data Mining an Ontology)
בשאלה זו תתנסו בחיפוש ישויות וקשרים באונטולוגיה DBpedia, עליה למדנו בהרצאה.
עליכם לתרגם את השאילתות מטה ל-SPARQL ולהריץ אותן ב:
https://dbpedia.org/sparql
חלק מהשאלות מכילות בסוגריים שמות ישויות ויחסים מ-DBpedia שעשויים לסייע בכתיבת השאילתא.
יש להגיש את השאילתא ואת **10** התוצאות הראשונות של הפלט בתוך קובץ הPDF.

א. שאילתא שמחזירה את שמות כל העיתונים שמתפרסמים בשפה הספרדית.

ב. שאילתא שמחזירה שחקני כדורגל שנולדו במדריד (resource/Madrid) ושיחקו בקבוצה (ontology/team) שהאצטדיון שלה (ontology/ground) ממוקם באנגליה (resource/England). החזירו את שם השחקן ואת שם הקבוצה.

ג. שאילתא שמחזירה נהרות (ontology/River) שעוברים גם בצרפת (resource/France) וגם בארצות (ontology/country) אחרות. החזירו את שם הנהר ואת הארץ בה הוא עובר.

ד. שאילתא על שחקני קולנוע שנולדו בארץ (ontology/country) שמדברים בה צרפתית (resource/French_language) וכיכבו (ontology/starring) בסרט מארץ שמדברים בה אנגלית (resource/English_language). החזירו את שם השחקן, את הארץ בה נולד, את שם הסרט ואת הארץ של הסרט.

שאלה 2: (IE generation of an Ontology)

בשאלה זאת נתמקד ב INFORMATION EXTRACTION מויקיפדיה.

(א)עליכם לכתוב תוכנית PYTHON בשם football_ontology.py שתיצור אונטולוגיה של קבוצות כדורגל ושחקני כדורגל שמשחקים בהן, כולל מקום ותאריך לידה, ותפקידם במגרש.

על האונטולוגיה שתוחזר להכיל את הקשרים הבאים:

?league	<country>	?country
?team	<league>	?league
?team	<homeCity>	?city
?player	<playsFor>	?team
?player	<birthPlace>	?city
?city	<located_in>	?country
?player	<birthDate>	?date
?player	<position>	?position

- התוכנית football_ontology.py תקבל כקלט כתובת url שמשויכת לדף הויקיפדיה של ליגת כדורגל, ספציפית של הליגה האנגלית בעונת 2019-2020:
https://en.wikipedia.org/wiki/2019%E2%80%9320_Premier_League
- על התוכנית לעבור על כל 20 הקבוצות (רמז: טבלת הקבוצות ב-HTML), ולכל אחת מהקבוצות לעבור על השחקנים שלהם (רמז: טבלת השחקנים).
- על התוכנית לייצר קובץ בשם ontology.nt שיכיל את האונטולוגיה שיצרה.

(ב) כתבו תוכנית בשם ontology_queries.py עבור האונטולוגיה שבניתם בסעיף (א). התוכנית תקבל כקלט קובץ ontology.nt של ליגת כדורגל ותריץ עליו, בעזרת RDFLIB, את השאילתות הבאות:

a. כל השחקנים שנולדו בברזיל ומשחקים בליגה ואת הקבוצה שלהם.
b. כל השחקנים שנולדו אחרי 1995 ואת הקבוצה שלהם.
c. שחקנים שנולדו בעיר בה שבה נמצאת קבוצתם.
d. כל משחקי הדרבי האפשריים בליגה (משחק של 2 קבוצות מאותה העיר).

(ג) הריצו את הקוד שכתבתם בסעיף (ב) על האונטולוגיה שבניתם בסעיף (א) והגישו את התוצאה. זכרו שראינו איך להריץ שאילתות SPARQL בעזרת פייתון.

<u>הערה</u>: על מנת לטעון תאריכים לאונטולוגיה יש להשתמש במחלקה Literal של ספריית rdflib, אשר מייצגת ערכים אטומיים דוגמת מחרוזת, מספר או תאריך (לעומת המחלקה URIRef המייצגת ישות באונטולוגיה).

דוגמה:

```
from rdflib import Literal, XSD
person = URIRef('http://example.org/Gal_Gadot')
relation = URIRef('http://example.org/birthDate')
dob = Literal('1985-04-30',datatype=XSD.date)
ontology.add((person, relation, dob))
```

שאלה 3:

(א) ממשו תוכנית CRAWLER בשפת PYTHON עבור אתר ויקיפדיה, שמקבלת כתובת URL של עמוד התחלתי, ובונה גרף של 10 ההצבעות **הפנימיות** (השונות) הראשונות בין כל העמודים, כלומר לינקים שמתחילים ב "****/wiki/". על ה CRAWLER להקפיד לא לבקר באותו עמוד פעמיים, כמו כן, יש לעצור אחרי עומק 3 צעדים מהעמוד המקורי, כאשר באיטרציה האחרונה יש לשמור רק את הלינקים לעמודים שכבר ביקרנו בהם (על מנת למנוע מצב של המון קודקודים ללא הצבעות החוצה).
זכרו שבמידה ולעמוד אין אף לינק פנימי שיוצא ממנו, אזי מדובר בבור ולכן עלינו להניח שקיימת לולאה עצמית בדף, כלומר הוא מצביע על עצמו.
יש להדפיס את הגרף בצורה של רשימות שכנות. לדוגמא:

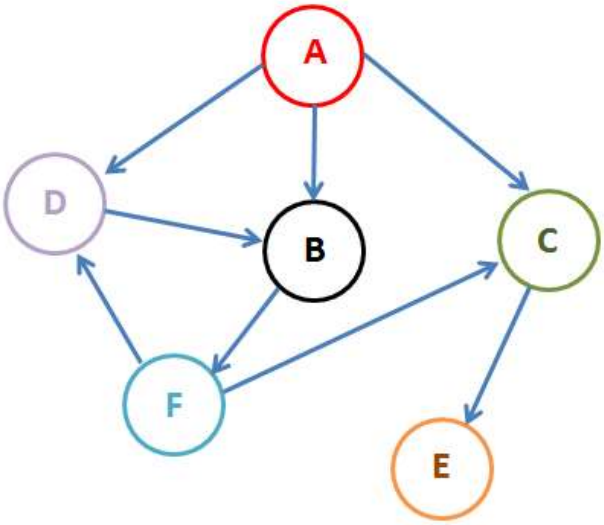
SiteA = {SiteB, SiteC, SiteD}
SiteB = {SiteA, SiteD}
SiteC = {SiteA, SiteD}

הגישו את התשובות בקובץ question3a.py

(ב) ממשו אלגוריתם PAGERANK, המקבל גרף מסעיף (א) ומחזיר את ערכי ה PR של העמודים. יש להשתמש ב DAMPING FACTOR של 0.3. על התוכנית להדפיס את ה PR ואת ה URL של כל אחד מהעמודים.
הגישו את התשובות בקובץ question3b.py

(ג) הריצו את התוכניות שפיתחתם בסעיפים (א) ו (ב) על האתר הבא:
https://en.wikipedia.org/wiki/Kirill_Nababkin
והדפיסו את התוצאות לקובץ question3c.txt

שאלה 4:



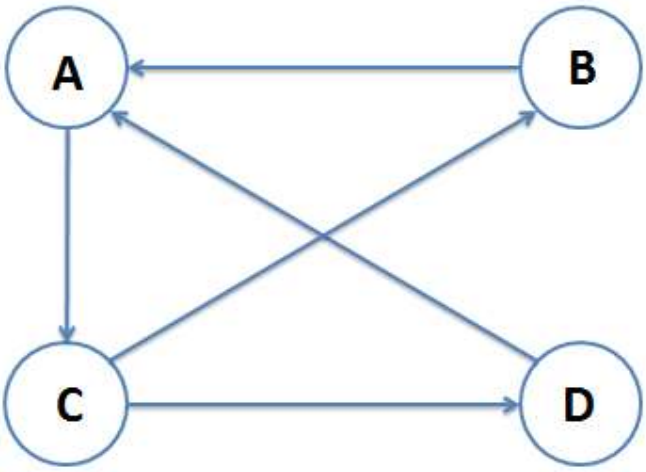
(א) נתון גרף של לינקים בין האתרים A, B, C, D, E, F. חשבו את ה PAGERANK של כל אחד מהאתרים ללא DAMPING FACTOR והסבירו את התוצאה שקיבלתם. הגישו את החישובים שלכם ואת ההסבר בתוך הקובץ answers.pdf

(ב) חשבו את ה PAGERANK של עם DAMPING FACTOR של 0.1 והגישו את החישובים בקובץ answers.pdf

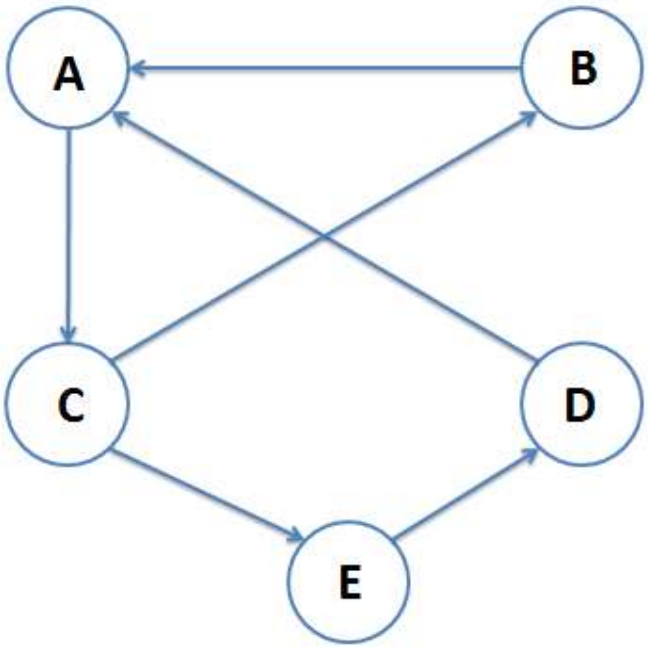
(ג) הסבירו את השוני בתוצאות של סעיף (א) וסעיף (ב) והגישו את ההסברים בקובץ answers.pdf

שאלה 5:

נתון גרף המייצג ארבעה דפי אינטרנט והקישורים ביניהם:



לאחר מכן, נוסף דף חדש (E), והקישורים התעדכנו. הגרף החדש מוצג להלן:



מה קרה לערכי ה PAGERANK של **כל אחד מהקודקודים** לאחר השינוי של מבנה הגרף? (הניחו שחישוב ה- PR נעשה תמיד ללא DAMPING FACTOR)
נמקו את תשובתכם והגישו את ההסבר בתוך הקובץ answers.pdf