

## Graph analysis and neo4j DB - הנחיות הגשה

עליכם להגיש סיכום של המעבדה **במסמך PDF או HTML אחד** כתוצר של מחברת ג'ופיטר.

המחברת תכלול קוד, טבלאות רלוונטיות, ויזואליזציות, את הסיכומים והתשובות ניתן לכתוב בכתב יד ולהוסיף כתמונות למסמך (בתנאי וקריא) או להקלידן. יש לתעד את הקוד באופן ברור עבור כל פעולה.  
יש לתת **כותרת ברורה וקריאה עבור כל גרף וטבלה** (ולא כותרת כללית לכלל הגרפים מאותו סוג)  
מקרא עבור הגרפים, כותרות לצירים (מומלץ גם משפט הסבר)  
במידה והתבקשתם להציג ערכים, בבקשה לכתוב במשפט את המסקנה מהגרף.  
תשובות לשאלות על ה DB יש לתת באותו מסמך, בנוסף יש לצרף צילומי מסך מההרצות שלכם.

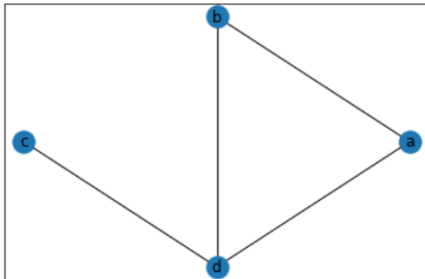
כנתונים לניתוח עליכם להשתמש בנתונים סינטטיים היוצרים **רשת קשרים באופן רנדומלי ועל 20 צמתים לפחות.**

דוגמה לבניית רשת אקראית וחישובי pagerank:

```
import itertools
import random
import networkx as nx
from matplotlib import pyplot as plt

nodes = ['a', 'b', 'c', 'd']
# Generate Networkx Graph
G = nx.Graph()
G.add_nodes_from(nodes)
# randomly determine vertices
for (node1, node2) in itertools.combinations(nodes, 2):
    if random.random() < 0.5:
        G.add_edge(node1, node2)
# Draw generated graph
nx.draw_networkx(G, pos=nx.circular_layout(G), with_labels=True)
# Compute Page Rank
pr = nx.pagerank(G, alpha=0.85)
plt.show()
print (pr)
```

→



```
{'a': 0.24592797270129024, 'b': 0.24592797270129024, 'c': 0.14140875554444027, 'd': 0.3667352990529791}
```

1. הסבירו וסכמו על LinkAnalysis במילים שלכם, ובהתייחס לנלמד במסלול.
2. סכמו והסבירו על האלגוריתמים HITS ו PageRank  
1. מתי לדעתכם נשתמש בכל אחד ? (למה הם מיועדים, ודוגמה לתרחיש שימוש)  
2. מהי צורת הייצוג לנתונים ?  
3. לאיזה תהליכי עיבוד מקדים נדרש ?  
4. יתרונות, חסרונות, סקלביליות זמני ריצה וכו'.
3. קראו והסבירו על מדדים אחרים למציאת צומת מרכזי

4. בעבור הספריות [/https://networkx.org](https://networkx.org) ו [/https://igraph.org/python](https://igraph.org/python)
1. קיראו על צורות הייצוג השונות שיש לנו לנתונים (סוגי גרפים - חד כיווני, דו כיווני, ממושקל וכו).
  2. קיראו על אלגוריתמי ה LinkAnalysis השונים. ( HITS ו PageRank ) התייחסו לנק הבאות תכונות, שיטות, פרמטרים, אופי הפעולה, וההפעלה ברמה הטכנית של כל אחד מהאלגוריתמים.
  3. קיראו על אפשרויות הוויזואליזציה, כולל דרכים להוסיף כותרות והסברים לצמתים וקשתות.
5. חקר ספריה לבחירתכם ברמה המעשית - בנו רשת אקראית של 20 צמתים.
1. בחלק זה עליכם להדגים את אופי העבודה של כל אחת מהחבילות ומציאת צמתים מרכזיים בעזרת HITS ו PageRank
  2. יש להציג זמני בניית צורות הייצוג, ועבודת האלגו'.
  3. יש להציג את הגדלים השונים של הגרפים שנוצרו (מבני הנתונים).
  4. יש להציג וויזואליזציות רלוונטיות של הגרפים.
6. סכמו את התוצאות של סעיף 4 באופן השוואתי וברור והסבירו במילים שלכם (יש להתייחס לנקודות הבאות):
- את התוצאות יש להציג גם בטבלה השוואתית בנוסף לוויזואליזציות מתאימות.
- i. ניתוח של אופי ייצוג הגרף.
  - ii. עבודה עם האלגוריתמים השונים.
  - iii. בדיקת זמני ריצה וגודל מבני נתונים רלוונטיים.
  - iv. מסקנות
7. בעזרת הספריה שלא נחקרה בסעיף 4 הציגו בעזרת הנתונים להלן את גרף הקשרים המשפחתיים וקשרי העבודה של משפחות האצולה של פירנצה מתקופת הרנסנס, כפי שנחקרו על ידי פדגייט וחבריו.
- נתחו את הנתונים, הציגו את המשפחות המרכזיות בכל תחום, והביעו את מסקנותיכם.

This example introduces the PageRank centrality measure with a network of Renaissance Florentine families around 1430. Specifically, we examine the PageRank centrality of the Florentine families in their marriage network. The families are nodes, and marriage ties between the families are edges in the network

[http://www.casos.cs.cmu.edu/computational\\_tools/datasets/external/padgett/index2.html](http://www.casos.cs.cmu.edu/computational_tools/datasets/external/padgett/index2.html)

8. קראו על neo4j
1. תיעוד [/https://neo4j.com/docs](https://neo4j.com/docs)
  2. כלי ענן - [/https://neo4j.com/sandbox](https://neo4j.com/sandbox)
3. Assume a graph consisting of students, courses, projects, rooms, and relationships between. The nodes and relationships are specified in Cypher (the declarative graph query language used by Neo4j), as given below.

Nodes:

```
CREATE (s1: Student {studentID: "1", lastName: "Doe", firstName: "Ana", middleName: "Maria"})
CREATE (s2: Student {studentID: "2", lastName: "Ung", firstName: "Peter", middleName: "John"})
```

```

CREATE (s3: Student {studentID: "3", lastName: "Doe", firstName: "John"})
CREATE (s4: Student {studentID: "4", lastName: "Berre", firstName: "Stine"})
CREATE (s5: Student {studentID: "5", lastName: "Travolta", firstName: "John"})
CREATE (c1: Course {courseNr: "1", courseName: "Databases"})
CREATE (c2: Course {courseNr: "2", courseName: "Programming"})
CREATE (c3: Course {courseNr: "3", courseName: "Graphics"})
CREATE (p1: Project {projectNr: "34", projectName: "eCommerce database"})
CREATE (p2: Project {projectNr: "24", projectName: "eCommerce website"})
CREATE (p3: Project {projectNr: "13", projectName: "User interface"})
CREATE (p4: Project {projectNr: "26", projectName: "Reporting"})
CREATE (r1: Room {roomName: "Pascal"})
CREATE (r2: Room {roomName: "Seminar C"})
CREATE (r3: Room {roomName: "Alpha"})
CREATE (r4: Room {roomName: "Beta"})

```

Relationships:

```

CREATE (c1) - [:TAKESPLACEIN] -> (r1)
CREATE (c1) - [:TAKESPLACEIN] -> (r3)
CREATE (c1) - [:TAKESPLACEIN] -> (r4)
CREATE (c2) - [:TAKESPLACEIN] -> (r2)
CREATE (s1) - [:ENROLLEDIN] -> (c1)
CREATE (s2) - [:ENROLLEDIN] -> (c1)
CREATE (s3) - [:ENROLLEDIN] -> (c2)
CREATE (s4) - [:ENROLLEDIN] -> (c1)
CREATE (s1) - [:WORKSON {hours: "1"}] -> (p1)
CREATE (s1) - [:WORKSON {hours: "2"}] -> (p2)
CREATE (s2) - [:WORKSON {hours: "3"}] -> (p1)
CREATE (s2) - [:WORKSON {hours: "4"}] -> (p2)
CREATE (s2) - [:WORKSON {hours: "1"}] -> (p3)
CREATE (s2) - [:WORKSON {hours: "1"}] -> (p4)
CREATE (s3) - [:WORKSON {hours: "1"}] -> (p1)
CREATE (s3) - [:WORKSON {hours: "2"}] -> (p2)
CREATE (s3) - [:WORKSON {hours: "3"}] -> (p4)

```

Specify the following queries in Cypher and execute them in Neo4j.

1. In which rooms does course with course number "1" take place in? Retrieve the course name and the names of the rooms in which the course takes place.
2. How many hours and in which projects does student with student number "1" works on? Retrieve the first name of the student, the project the student works on and the corresponding number of hours worked on the project.
3. Which students and how many hours do they work on the project with project number "24"? Retrieve the project name, the last name of the student and the corresponding number of hours worked on the project.
4. Which students work in which projects and how many hours? Retrieve the last name of the students, the name of the projects they work on, and the corresponding number of hours. Order the results by the last name of the students. Limit the results to four.

5. Which students work on more than two projects and on how many projects exactly?  
Retrieve the last name of the students and the corresponding number of projects. Order the results by the number of projects.
6. Which students have the same last name and work on the same projects? Retrieve the first name of the students and the name of projects they share.

**Basic/Simplified syntax of some common Cypher clauses:**

- Create nodes  
*CREATE <node, optional labels and properties>*
- Create relationships  
*CREATE <relationship, relationship type and optional properties>*
- Delete nodes or relationships:  
*DELETE <nodes or relationships>*
- Find nodes and relationships that match a pattern  
*MATCH <pattern>*
- Specify aggregates and other query variables  
*WITH <specifications>*
- Specify conditions on the data to be retrieved  
*WHERE <condition>*
- Specify data to be returned  
*RETURN <data>*
- Order the data to be returned  
*ORDER BY <data>*
- Limit the number of returned data items  
*LIMIT <max number>*

9. קראו על neo4j-nx - ספריה זו מאפשרת להריץ את היכולות של networkx על neo4j

1. <https://www.kaggle.com/yusufbaktir/nxneo4j-tutorial>

2. נתונים שני קבצי CSV

i. Asoiaf-book1-edges

ii. Asoiaf-book1-nodes

3. יבאו את הנתונים ל NEO4J

4. הציגו ניתוח של הקשרים בעזרת neo4j-nx

5. הציגו את הקשרים באופן ויזואלי

10. הוסיפו או שנו את האפליקציה שנכתבה עבור מונגו DB במעבדה 1

1. יש להוסיף יכולות לניתוח ושמירת גרפים והצגת וויזואליזציות שלהם

2. הורידו נתונים בעלי קשר מרשת חברתית (לפחות 20 צמתים) לבחירתכם

3. הציגו ניתוח של הקשרים

4. הציגו את הקשרים באופן וויזואלי (ניתן לשמור את התוצר הוויזואלי כקובץ - אין צורך להציג כ GUI)

בהצלחה !