19 פרויקט: חישוב ומציאת תתי-רשתות

בהכנה לפרויקט הבא שלנו שמאתר מסלולים ברשת, עלינו לבצע עבודה בהבנת כיצד כתובות IP, מסכות תת-רשתות ותתי-רשתות פועלות יחד.

בפרויקט זה ניישם חלק מהעבודה מהפרקים בפועל. אנחנו נ:

- נכתוב פונקציות להמיר כתובות IP בפורמט נקודות ומספרים לערכים בודדים של 32 סיביות – ולהפך.
- נכתוב פונקציה שממירה מסכת תת-רשת בגרסה עם סלש (slash notation) לערך בודד של 32 סיביות שמייצג את המסכה.
- נכתוב פונקציה שתבדוק אם שתי כתובות IP נמצאות באותה תת-רשת.

19.1 מגבלות

אסור להשתמש ב:

- כל פונקציה ממודול ה-socket.
- כל פונקציה ממודול ה-struct.
- כל פונקציה ממודול ה-netaddr.
- שיטות .to_bytes() או .from_bytes().
- יש להסתפק בעיבודים ביטיים שבניתם בעצמכם.

19.2 מה לעשות

הורד את הקוד הבסיסי והקבצים האחרים מקובץ ה-ZIP הזה. אלו הם הקבצים שתגמור למלא בפרויקט הזה.

מימוש הפונקציות הבאות בקובץ `netfuncs.py`:

python
Copy code
```
(ipv4_to_value(ipv4_addr
(value_to_ipv4(addr
(get_subnet_mask_value(slash
(ips_same_subnet(ip1, ip2, slash
(get_network(ip_value, netmask
(find_router_for_ip(routers, ip
```

ההסברים על הפונקציות נמצאים בקובץ בתוך ה-docstrings שלהן. שים לב במיוחד לסוגי הקלט והפלט בדוגמאות שמופיעות שם.

שימו לב כי אף אחת מהפונקציות לא צריכה להיות ארוכה מ-5-15 שורות. אם אתם מקבלים מימוש פונקציה הרבה יותר ארוך, ייתכן שאתם לא בכיוון הנכון.

19.3 בדיקות תוך כדי עבודה

אני ממליץ שתכתוב פונקציה אחת בכל פעם ותבדוק אותה על ידי קריאתה עם נתונים לדוגמה לפני שתעבור לפונקציה הבאה.

תוכל להוסיף קריאות לפונקציות שלך כדי לוודא שהן עושות את מה שהן אמורות לעשות. השתמש בקלטים ובפלטים שמופיעים בהערות הדוגמאות לצורך בדיקות.

יש פונקציה בשם `my_tests()` בקובץ `netfuncs.py` שתורץ במקום פונקציית `main` ברירת המחדל אם תבצע לה un-comment.

אם תבצע un-comment לפונקציה `my_tests()`, תוכל להריץ את התוכנית כך:

bash
Copy code
```
python netfuncs.py
```

ותראה את הפלט של הפונקציה הזו.

ודא שתבצע comment לפונקציה `my_tests()` ותהריץ את התוכנית עם הקוד הראשי הכלול לפני שתשלח את העבודה, כפי שמופיע בסעיף הבא.

19.4 הרצת התוכנית
תוכל להריץ אותה כך:

bash
Copy code
```
python netfuncs.py example1.json
```

היא תקרא את נתוני ה-JSON מקובץ `example1.json` הכלול ותריץ את הפונקציות שלך על חלקים שונים ממנו.

הפלט, הכלול בקובץ `example1_output.txt`, אמור להיראות בדיוק כך אם הכל עובד כראוי:

yaml
Copy code
```
Routers:
  netmask 255.255.255.0: network 10.34.166.0 :10.34.166.1
  netmask 255.255.255.0: network 10.34.194.0 :10.34.194.1
  netmask 255.255.255.0: network 10.34.209.0 :10.34.209.1
  netmask 255.255.255.0: network 10.34.250.0 :10.34.250.1
   netmask 255.255.255.0: network 10.34.46.0 :10.34.46.1
   netmask 255.255.255.0: network 10.34.52.0 :10.34.52.1
   netmask 255.255.255.0: network 10.34.53.0 :10.34.53.1
   netmask 255.255.255.0: network 10.34.79.0 :10.34.79.1
   netmask 255.255.255.0: network 10.34.91.0 :10.34.91.1
   netmask 255.255.255.0: network 10.34.98.0 :10.34.98.1

IP Pairs:
```

```
different subnets :10.34.91.252    10.34.194.188
different subnets :10.34.91.120    10.34.209.189
different subnets :10.34.166.26    10.34.209.229
different subnets :10.34.91.184    10.34.250.213
different subnets :10.34.52.119    10.34.250.228
different subnets :10.34.46.73     10.34.250.234
different subnets :10.34.166.228   10.34.46.25
different subnets :10.34.91.55     10.34.52.118
different subnets :10.34.166.1     10.34.52.158
      same subnet :10.34.52.244    10.34.52.187
different subnets :10.34.46.130    10.34.52.23
different subnets :10.34.46.125    10.34.52.60
      same subnet :10.34.79.58     10.34.79.218
different subnets :10.34.46.142    10.34.79.81
different subnets :10.34.46.205    10.34.79.99
different subnets :10.34.53.190    10.34.91.205
different subnets :10.34.79.122    10.34.91.68
different subnets :10.34.46.255    10.34.91.97
different subnets :10.34.209.6     10.34.98.184
different subnets :10.34.166.170   10.34.98.33

                                   :Routers and corresponding IPs
,'10.34.166.228' ,'10.34.166.170' ,'10.34.166.1'] :10.34.166.1
                                                  ['10.34.166.26'
                              ['10.34.194.188'] :10.34.194.1
['10.34.209.6' ,'10.34.209.229' ,'10.34.209.189'] :10.34.209.1
               ,'10.34.250.228' ,'10.34.250.213'] :10.34.250.1
                                                  ['10.34.250.234'
 ,'10.34.46.142' ,'10.34.46.130' ,'10.34.46.125'] :10.34.46.1
      ['10.34.46.73' ,'10.34.46.255' ,'10.34.46.25' ,'10.34.46.205'
 ,'10.34.52.158' ,'10.34.52.119' ,'10.34.52.118'] :10.34.52.1
      ['10.34.52.60' ,'10.34.52.244' ,'10.34.52.23' ,'10.34.52.187'
                               ['10.34.53.190'] :10.34.53.1
  ,'10.34.79.58' ,'10.34.79.218' ,'10.34.79.122'] :10.34.79.1
                                    ['10.34.79.99' ,'10.34.79.81'
 ,'10.34.91.205' ,'10.34.91.184' ,'10.34.91.120'] :10.34.91.1
       ['10.34.91.97' ,'10.34.91.68' ,'10.34.91.55' ,'10.34.91.252'
                  ['10.34.98.33' ,'10.34.98.184'] :10.34.98.1
```

אם אתה מקבל פלט שונה, נסה לבדוק את הקוד ולראות אילו פונקציות משמשות עם הפלט השגוי. אז תוכל לבדוק אותן בפרטים יותר בקפידה בפונקציה `my_tests()`.