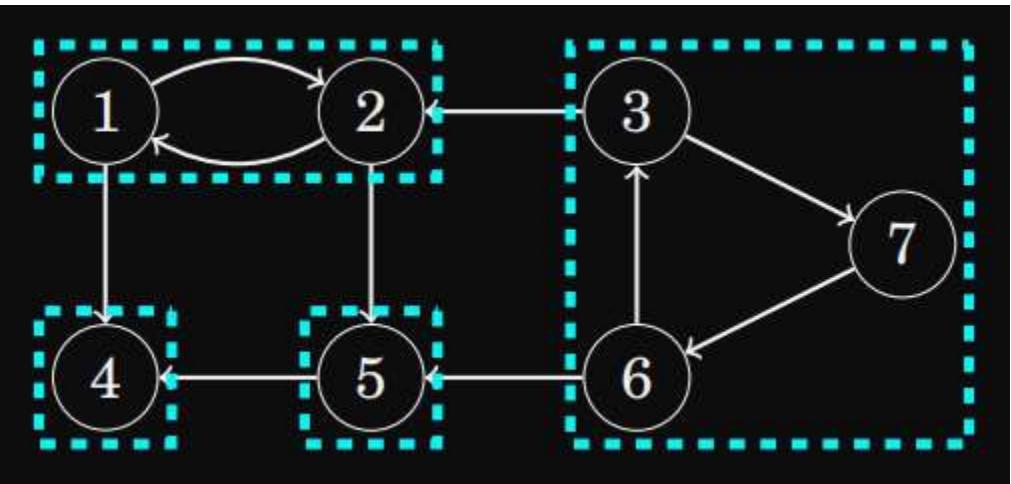


**גרפים -2** UF/RB and SCC

## SCC - קשריות חזקה

- איך מוגדרת קשריות בגרף מכוון? אם ט. יכול להגיע לא-ז האם זה אומר שהם חלק מאותו רכיב קשריות?
- התשובה היא שבגרפים מכוונים **קשריות נמדדת לפי קשריות חזקה**. שני צמתים ט, ו הם **strongly connected** אם ומ"מ ניתן להגיע מ-ט לא-ז, וגם ניתן להגיע מ-א-ז ל-ט.



- איך נמצא את רכיבי הקשריות חזקה בגרף?  
בහינתן צמתים ט, ו איך נדע האם הם  
באותו רכיב קשריות חזקה?

# אלגוריתם קשירותה חזקה

1. נעשה  $\text{dfs}$  החל מכל צומת, וכל פעם ראשונה שעוברים בצומת נדחוף אותו  $\text{stack}$  ברגע שנסויים את  $\text{dfs}$  ממנה (עברנו כבר על כל שכניו).
2. נהפוך את הכוון של כל קשת בגרף.
3. נעשה  $\text{dfs}$  החל מכל צומת לפי הסדר שלהם ב $\text{stack}$ . יהיה לנו מערך שזכור לכל צומת את מספר רכיב השירות שלו, כל פעם שנתחילה  $\text{dfs}$  מצומת חדש ב $\text{stack}$  אנחנו נתחיל רכיב שירות חדש, וכל פעם שנגיע לצומת שעוד לא הגיעו אליו נזכיר לו את רכיב השירות הנוכחי.

# הוכחה לאלגוריתם לקשרות חזקה

בשביל עוד מידע ובשביל הוכחה של האלגוריתם, מומלץ לחפש בגוגל  
ולסתכל כאן

<https://cp-algorithms.com/graph/strongly-connected-components.html>

# Union find

- באך נרצה עבור גרפים אלגוריתם שזכור קבוצות ויעבוד כר:
- נוכל לחבר בין שתי קבוצות ב  $O(\alpha)$ .
- בהינתן מישהו נוכל לדעת באיזו קבוצה הוא ב  $O(\alpha)$ .

# כולם מכירים את הרעיון - נראה את הקוד

```
44
23     const int maxn = 1e5+5;
24     int p[maxn] = {0}, sz[maxn] = {0}; // technically redundant.
25     void pre(){ rep(i, 0, maxn) p[i] = i, sz[i] = 1; }
26     inline int find(int a) { return a == p[a] ? a : p[a] = find( a: p[a]); }
27     int onion(int a, int b){
28         if((a = find(a)) == (b = find( a: b))) return 0;
29         if(sz[a] < sz[b]) swap( & a, & b);
30         return sz[a] += sz[b], p[b] = a, 1;
31     }
```