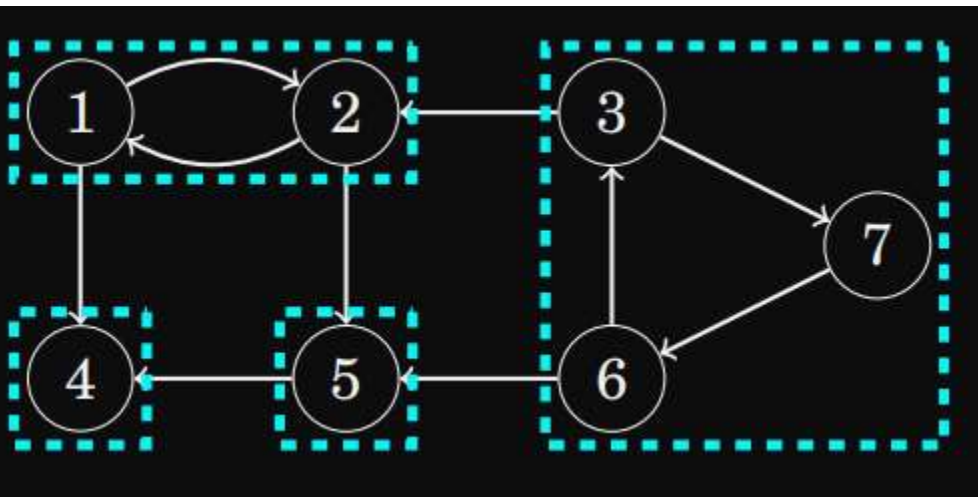


גרפים -2 UF/RB and SCC

SCC - קשירות חזקה

- איך מוגדרת קשירות בגרף מכוון? אם u יכול להגיע ל- v האם זה אומר שהם חלק מאותו רכיב קשירות?

- התשובה היא שבגרפים מכוונים קשירות נמדדת לפי קשירות חזקה. שני צמתים u, v הם *strongly connected* אם ניתן להגיע מ- u ל- v , וגם ניתן להגיע מ- v ל- u .



- איך נמצא את רכיבי הקשירות החזקה בגרף? בהינתן צמתים u, v איך נדע האם הם באותו רכיב קשירות חזקה?

אלגוריתם קשירות חזקה

1. נעשה dfs החל מכל צומת, וכל פעם ראשונה שעוברים בצומת נדחוף אותו לstack ברגע שנסיים את dfs ממנו (עברנו כבר על כל שכניו).
2. נהפוך את הכיוון של כל קשת בגרף.
3. נעשה dfs החל מכל צומת לפי הסדר שלהם בstack. יהיה לנו מערך שזוכר לכל צומת את מספר רכיב הקשירות שלו, כל פעם שנתחיל dfs מצומת חדש בstack אנחנו נתחיל רכיב קשירות חדש, וכל פעם שנגיע לצומת שעוד לא הגענו אליו נזכור לו את רכיב הקשירות הנוכחי.

הוכחה לאלגוריתם לקשירות חזקה

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

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

Union find

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

כולכם מכירים את הרעיון - נראה את הקוד

```
22
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 union(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  }
```