<u>ייצוג 3:</u> שילוב של שני הייצוגים הקודמים.
מערך של שמות הקבוצות + רשימות מקושרות של הקבוצות.
כדי לבצע find **הצמתים ימומשו במערך.**

	A	next
1	1	3
2	3	6
3	1	0
4	3	7
5	2	0
6	3	4
7	3	0

	first
S_1	1
S_2	5
S_3	2

צומת ראשון של כל קבוצה

find(i) **יבוצע ע"י** A[i]

Union(p,q) : **אחד את הקבוצה המתחילה ב-**first[p]
עם זו המתחילה ב-first[q].

בזמן האיחוד צריך לעדכן את A -- **זמן** $O(n)$.

<u>משאלה:</u>

בכל איחוד

מוסיפים את הקבוצה הקטנה לגדולה,

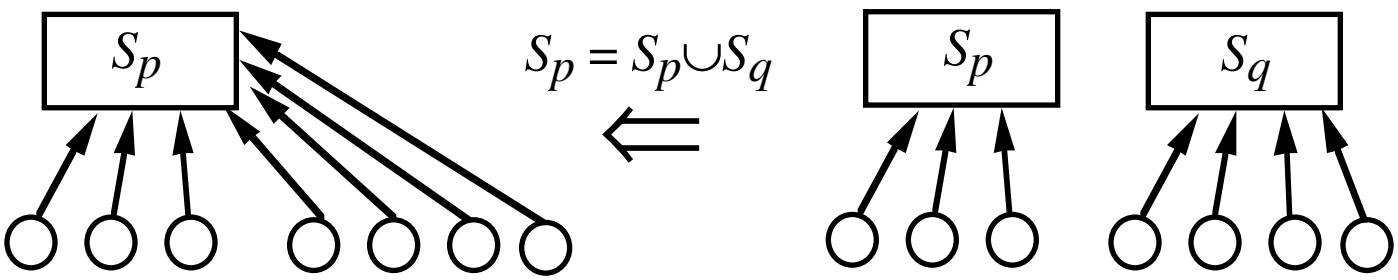
כלומר, כשמבצעים

$S_p = S_p \cup S_q$, מתקיים $|S_p| \geq |S_q|$.

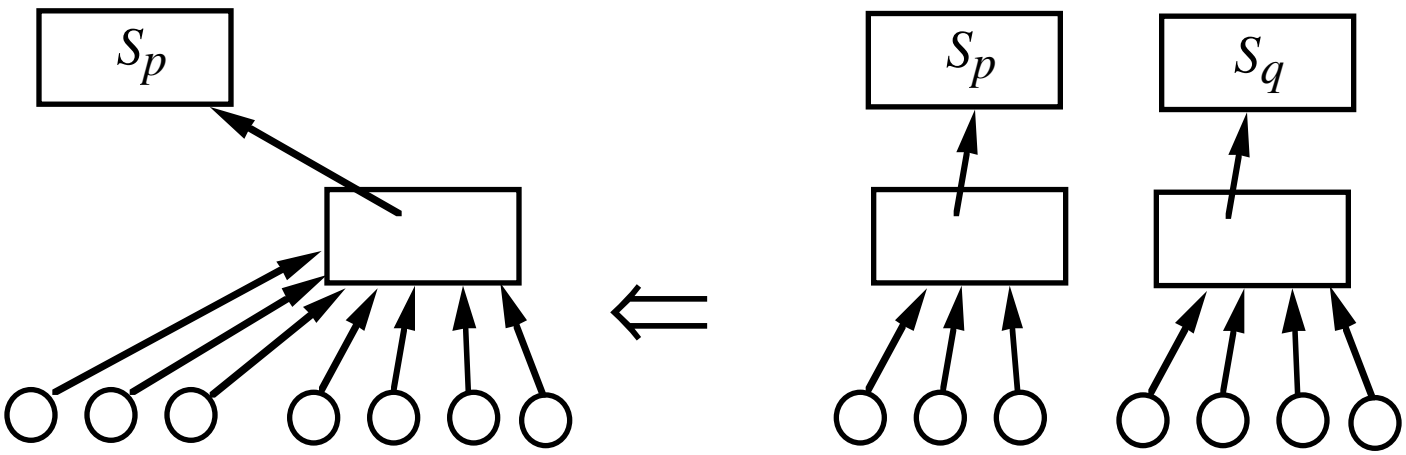
<u>טענה:</u> אם מתקיימת המשאלה, אזי כל איבר x משנה את הקבוצה אליה הוא שייך לכל היותר $\log n$ פעמים.

<u>מסקנה:</u> אם מתקיימת המשאלה, ומתחילים עם קבוצות יחידה, סדרה כלשהי של m פעולות Union תיקח לכל היותר זמן $O(m \log n)$.

כדי למלא את המשאלה



נוסיף עוד רמה של מצביעים:



מימוש 4:

בכל צומת נוסיף שדה p_name
ובנוסף יהיה מערך real_name.

פעולת find(v):

```
Find(v){
    return  real_name[p_name[v]];
}
```

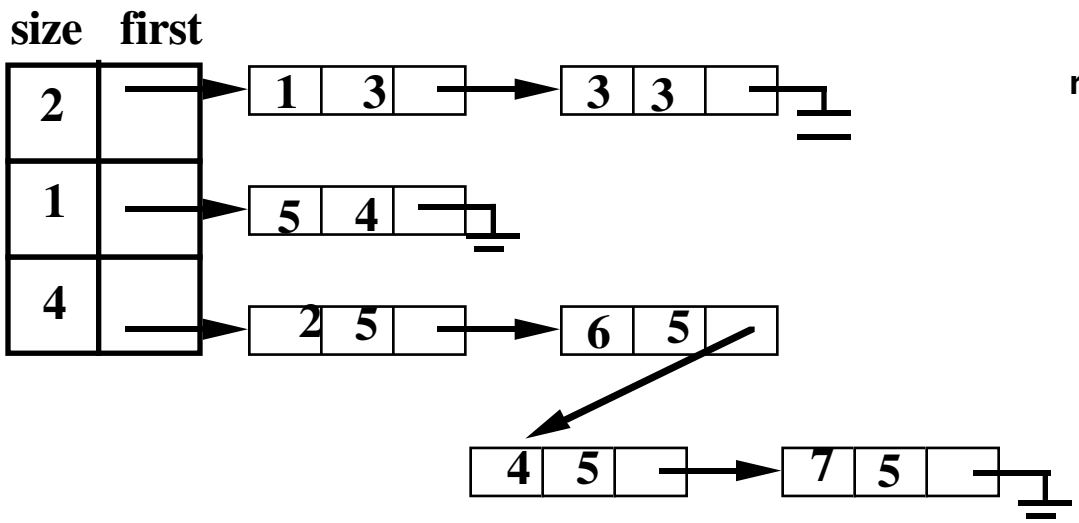
דוגמא: הקבוצות: $S_1 = \{1,3\}$ $S_2 = \{5\}$ $S_3 = \{2,4,6,7\}$

	p_name	next
1	3	3
2	5	6
3	3	0
4	5	7
5	4	0
6	5	4
7	5	0

	size	first
S_1	2	1
S_2	1	5
S_3	4	2

	real_name
3	1
4	2
5	3

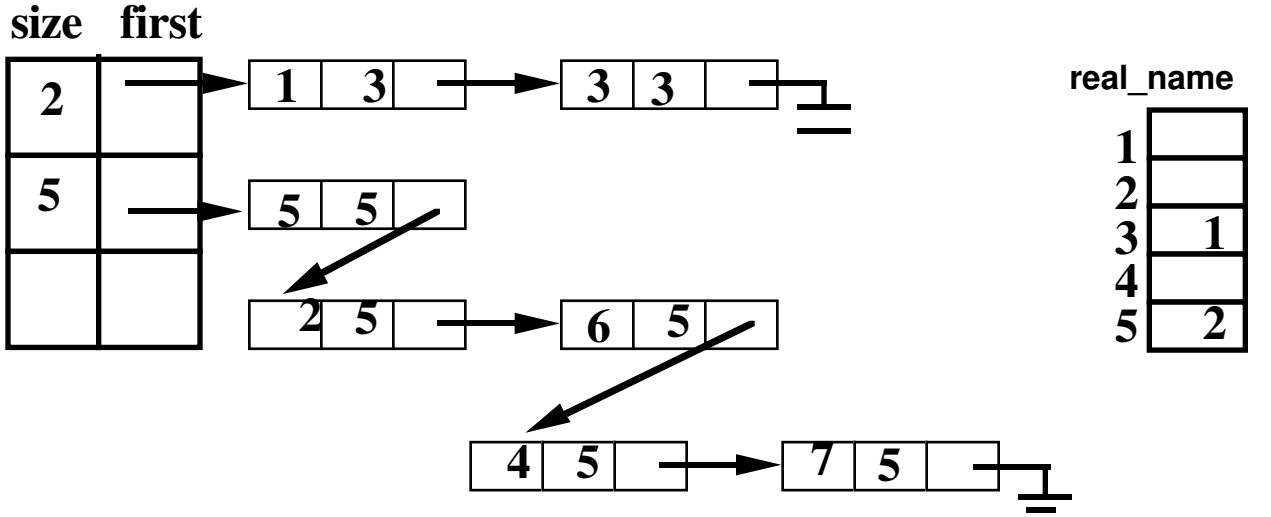
באופן גרפי:



באיחוד, נצרף את הקבוצה הקטנה לגדולה, ונעדכן את real_name.

```
Union(p,q){
    if(size[p]≤size[q]){
        small=p;  psuedo=p_name[first[q]];
    }
    else{
        small=q;  psuedo=p_name[first[p]];
    }
    for(v=first[small];  v≠0;  v=next[v])
        p_name[v]=pseudo;
    real_name[pseudo]=p;
}
```

דוגמא: אחרי ביצוע $S_2 = S_2 \cup S_3$:



סיבוכיות

find זמן $O(1)$,

m פעולות Union, זמן $O(m \log n)$.

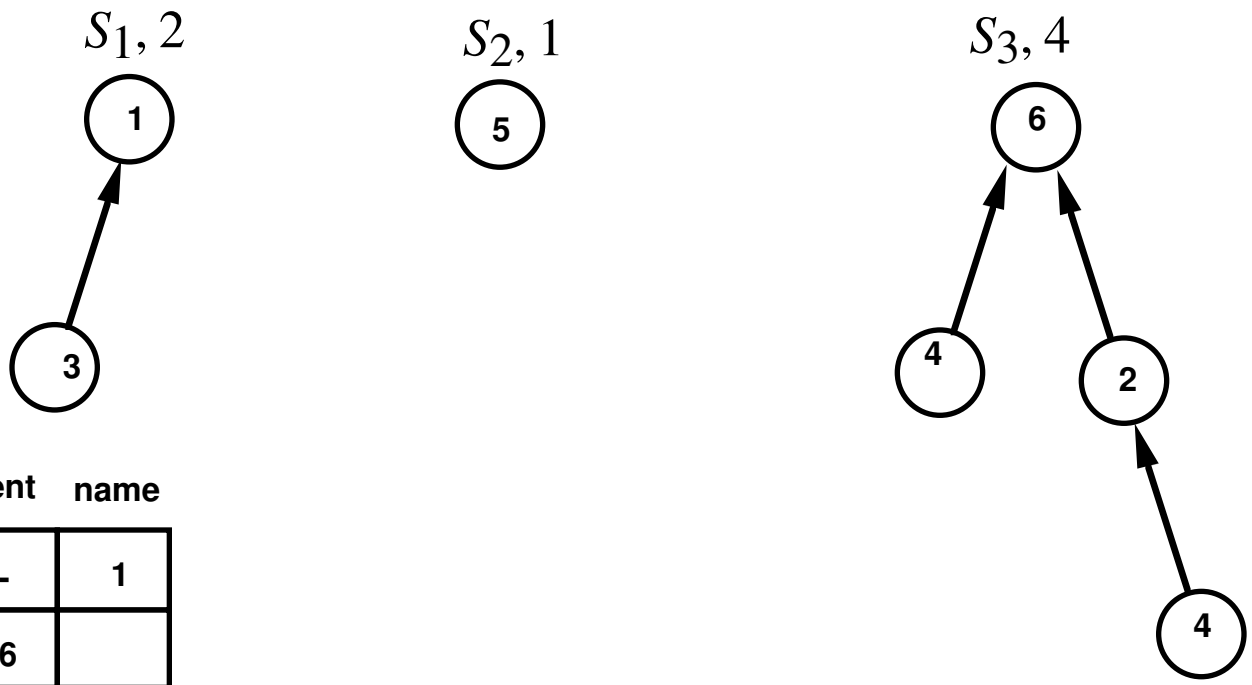
נאמר שפעולת Union לוקחת זמן משוערך $O(\log n)$ (amortized time).

ייצוג 5: עצים הפוכים Up Tree

כל קבוצה תיוצג ע"י עץ.

כל צומת יכיל מצביע לאביו (אין מצביעים לבנים).

שורש העץ יכיל את שם הקבוצה וגודלה.



	parent	name
1	-	1
2	6	
3	1	
4	6	
5	-	2
6	-	3
7	2	

	root	size
S_1	1	2
S_2	5	1
S_3	6	4

הקבוצות

Find(v): טפס בעץ מ-v עד לשורש,

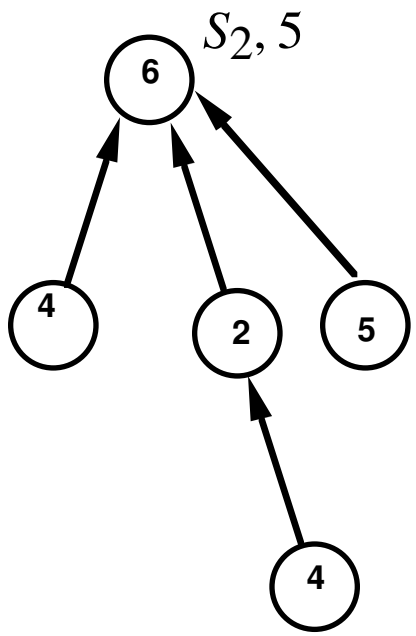
ושם תמצא את שם הקבוצה.

Union(p,q): אחד את העצים של S_p ו-S_q
ע"י זה שהופכים את שורש הקבוצה הקטנה להיות בן של שורש הקבוצה הגדולה.

דוגמה: אם נבצע $S_2 = S_2 \cup S_3$:

	parent	name
1	-	1
2	6	
3	1	
4	6	
5	6	
6	-	3
7	2	

	root	size
S_1	1	2
S_2	6	5
S_3	-	



סיבוכיות זמן:
O(1) -- Union
Find -- גובה העץ.

טענה: לעץ בעל גובה h, יש לפחות 2^h צמתים.
הוכחה: אינדוקציה על h.

מסקנה: ביער בן n צמתים, גובה כל עץ מקיים $h \le \log_2 n$.

מכאן: זמן פעולת Find הוא $O(\log n)$.