Implementing the G-S Algorithm. (n men and n women).

- Maintain a list of free men.

removal: m proposes and is accepted (delete m)

insertion, w breaks engagement m (insert m)

query : take an element m from this list.

Linked List!

O(1) time
1. When the algorithm asks for a free man, return the first element. m.
2. If m becomes engaged, remove m.
3. If m' becomes free, add m' to beginning of the Linked List.

Q: Suppose in step 1, we have to give the element in the middle.
This takes $\Theta(s)$ time [s: current # of free men]

Q: Suppose we use an array for 2. this takes $\Theta(n)$ time.

Q: Can we store the list of free men using an array?

Yes!
1. When algorithm asks for a free man, return the last element m in the array.
2. (possibly) delete m.
3. (possibly) add m' to the end of the array.

- Identify the highest-ranked woman on m's list that m hasn't proposed to.

Example. m_3 : $w_3 > w_1 > w_2 > w_4$.

Using Array : m_3 [3 | 1 | 2 | 4] next[m_3] = 0.

index 0 1 2 3

```
Next-woman (int m)
    w = man-pref[m][next[m]];
    next[m] = next[m] + 1;
return w.
```

Man-pref is a 2-D array:

1	2	3	4
2	3	1	4
3	1	2	4
1	2	3	4

(from HW1):
Man-pref[i][j] stores the index of m_i's j-th favorite woman.

- Next woman in w's preference (cont.)

  It is possible to do it with a linked list.

  Maintaing a counter doesn't work for linked list.

  Possible solutions: delete a woman from m's preference after m proposes to her.

  Alternatively: Maintain pointers.

- Who is w's current partner? (Is w free?)

  Array: Current[w] stores the index of w's current partner

  (it is set to NULL or -1) if w is free.

- w: $m > m'$?

  Attempt: go over w's preference and check if m appears before m'.

  $\Theta(n)$ time. (slow!)

  Can be done in $O(1)$ time!

  Ranking[w][m] returns m's position on w's preference list.

(Example)

w_1: $m_2 > m_3 > m_4 > m_1$

Ranking[w_1] =

m_1	m_2	m_3	m_4
4th	1st	2nd	3rd

$\Downarrow$

m_1 is ranked 4th on w_1's list.

w: $m > m'$? $\iff$ Ranking[w][m] < Ranking[w][m']

(Example)

w_2: $m_3 > m_4 > m_1 > m_2$

m_1	m_2	m_3	m_4
3rd	4	1	2

w_2: $m_4 > m_1$?

Yes, because Ranking[w_2][m_4] = 2 < 3 = Ranking[w_2][m_1]