Disjoint Sets

Data Structures 2013: Lectured by Dr. Min-Zheng Shieh @ I^{2}CRC, NCTU.

ADT: Disjoint Sets

- Objects: a partition of {1,2,...,n}.
  - For example: ({1,2,3},{4,5,6},{7,8,...,n})
- Operations:
  - Union(x,y): union X and Y where $x \in X$, $y \in Y$.
  - Find(x): return the representative of X containing x.
  - SameSet(x,y): return if x and y are in the same set.
  - SetSize(x): return |X| where $x \in X$.

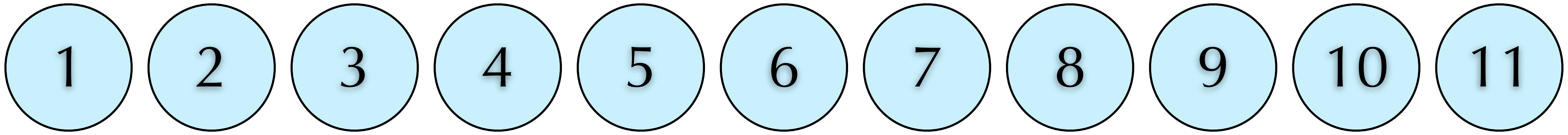
Disjoint sets

- Initially, the partition is ({1},{2},...,{n})
- Concept:
  - Representative of x: an element in X such that x∈X.
- Structure: Tree-like
  - Note: The parents don’t know who’s its child, but every child knows its parent.
- Implementation: Array

Initialization

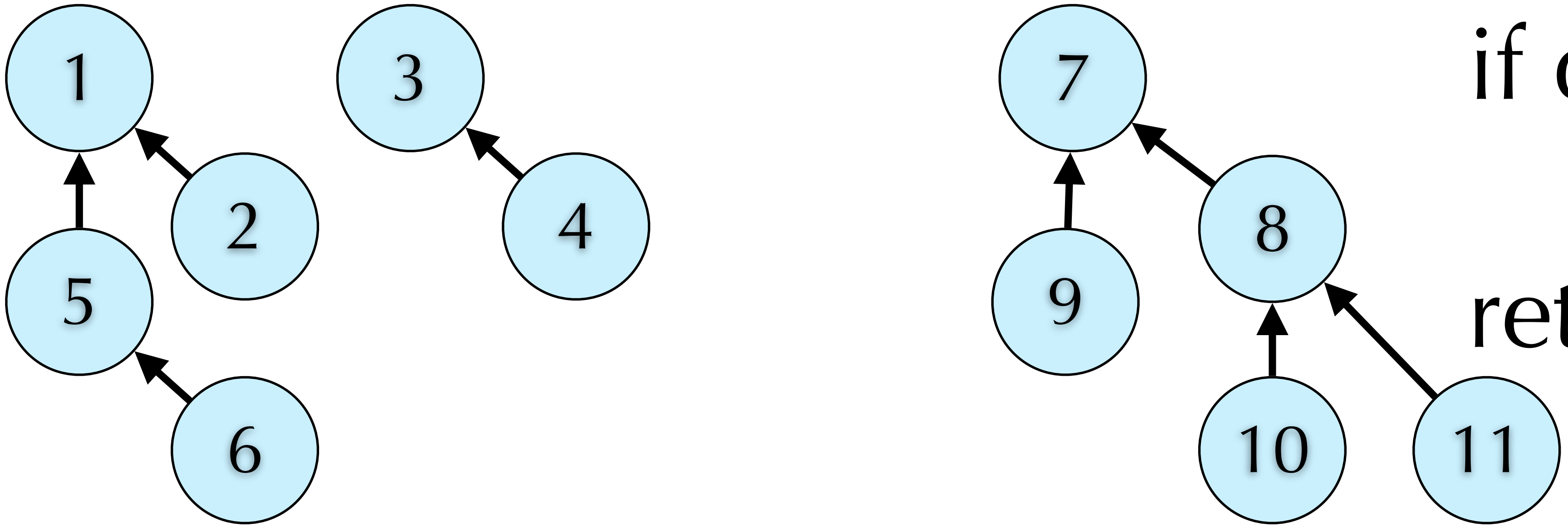
Representative: r_i

i	1	2	3	4	5	6	7	8	9	10	11
r_i	1	2	3	4	5	6	7	8	9	10	11
q_i	1	2	3	4	5	6	7	8	9	10	11



Find

i	1	2	3	4	5	6	7	8	9	10	11
r_i	1	1	3	3	1	1	7	7	7	7	7
q_i	1	1	3	3	1	5	7	8	7	8	8

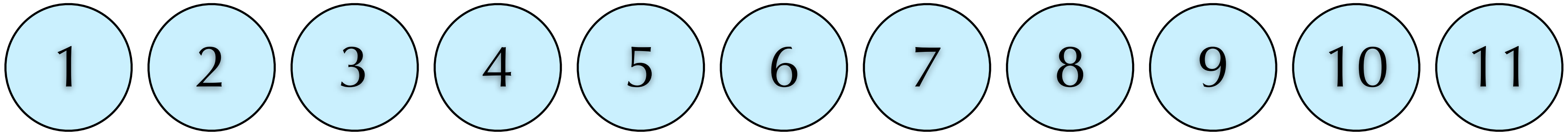


```
Find(x):
if $q_x$=x then
    return x;
return Find($q_x$);
```

Union

i	1	2	3	4	5	6	7	8	9	10	11
r_i	1	2	3	4	5	6	7	8	9	10	11
q_i	1	2	3	4	5	6	7	8	9	10	11

Union(1,2)

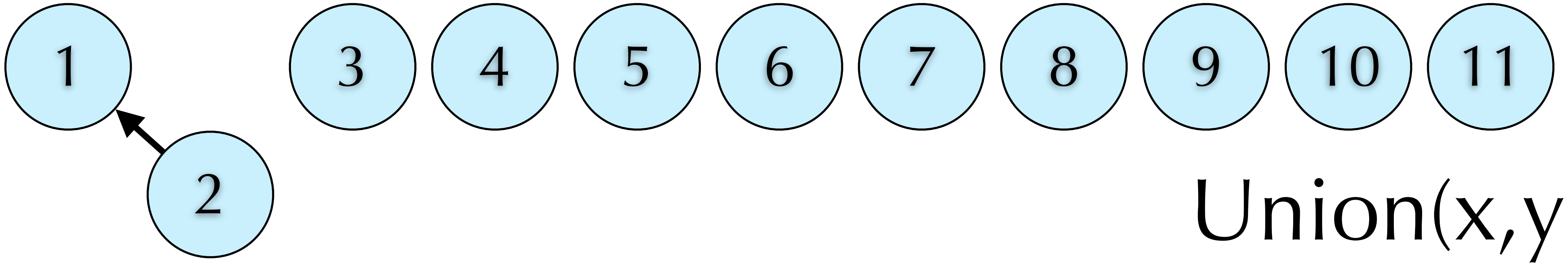


```
Union(x,y):
a=Find(x);
b=Find(y);
if $a \neq b$ then
   $q_b$=a;
```

Union

i	1	2	3	4	5	6	7	8	9	10	11
r_i	1	1	3	4	5	6	7	8	9	10	11
q_i	1	1	3	4	5	6	7	8	9	10	11

Union(3,4)

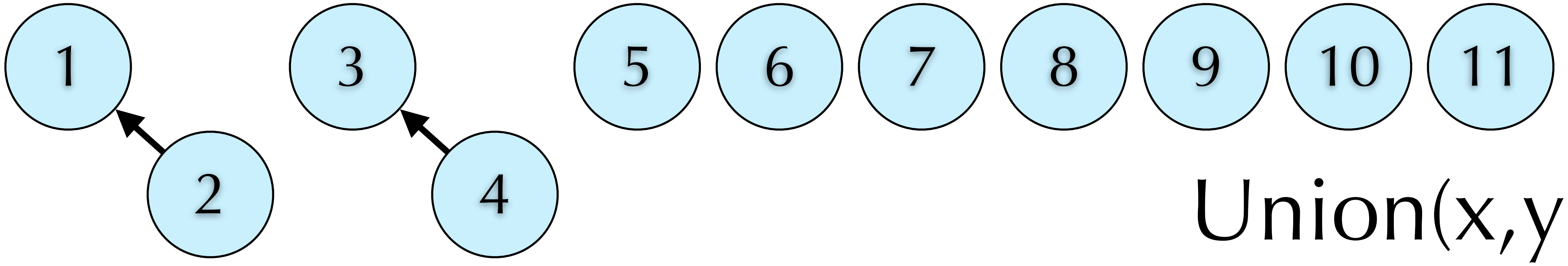


```
Union(x,y):
a=Find(x);
b=Find(y);
if $a \neq b$ then
    $q_b$=a;
```

Union

i	1	2	3	4	5	6	7	8	9	10	11
r_i	1	1	3	3	5	6	7	8	9	10	11
q_i	1	1	3	3	5	6	7	8	9	10	11

Union(5,6)

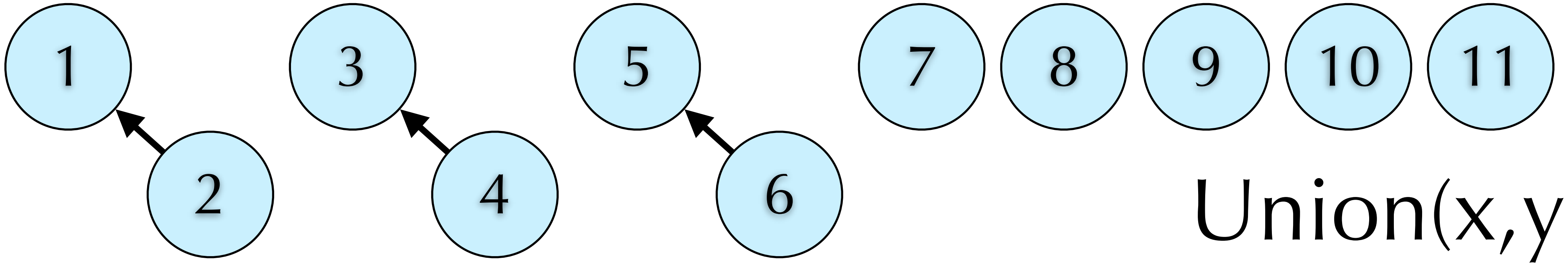


```
Union(x,y):
a=Find(x);
b=Find(y);
if a≠b then
    $q_b$=a;
```

Union

i	1	2	3	4	5	6	7	8	9	10	11
r_i	1	1	3	3	5	5	7	8	9	10	11
q_i	1	1	3	3	5	5	7	8	9	10	11

Union(2,6)

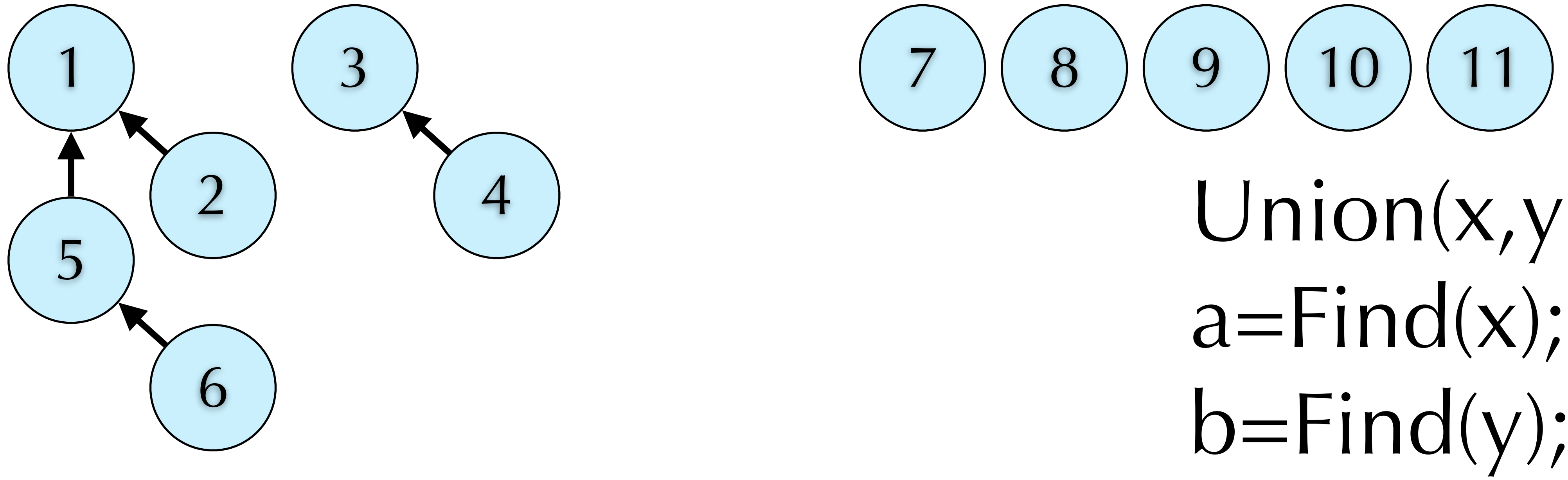


```
Union(x,y):
a=Find(x);
b=Find(y);
if a≠b then
    $q_b$=a;
```

Union

i	1	2	3	4	5	6	7	8	9	10	11
r_i	1	1	3	3	1	1	7	8	9	10	11
q_i	1	1	3	3	1	5	7	8	9	10	11

(why not 2?)

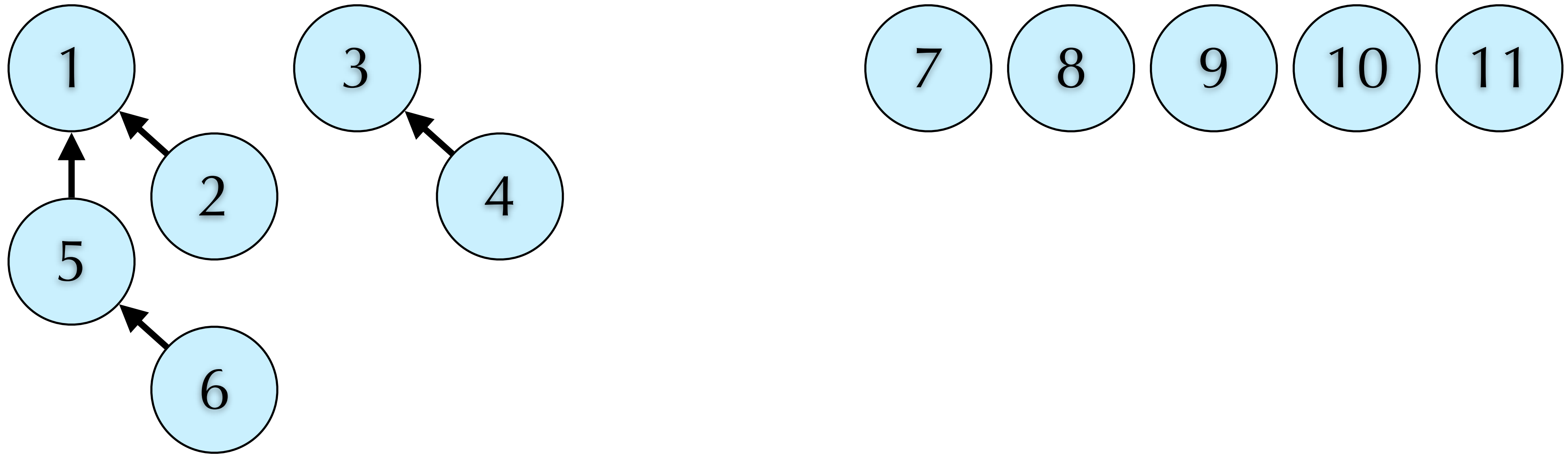


```
Union(x,y):
a=Find(x);
b=Find(y);
if $a \neq b$ then
   $q_b$=a;
```

Find w/ compression

i	1	2	3	4	5	6	7	8	9	10	11
r_i	1	1	3	3	1	1	7	8	9	10	11
q_i	1	1	3	3	1	5	7	8	9	10	11

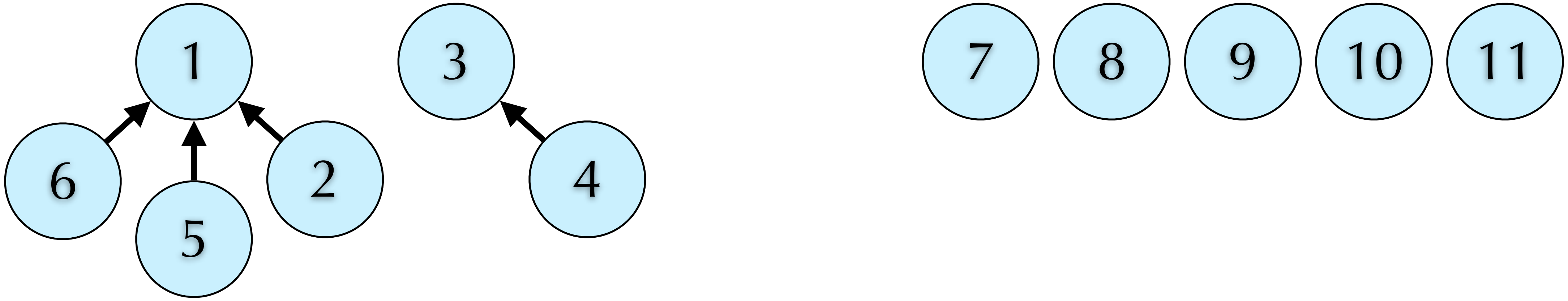
find(6)



Find w/ compression

i	1	2	3	4	5	6	7	8	9	10	11
r_i	1	1	3	3	1	1	7	8	9	10	11
q_i	1	1	3	3	1	1	7	8	9	10	11

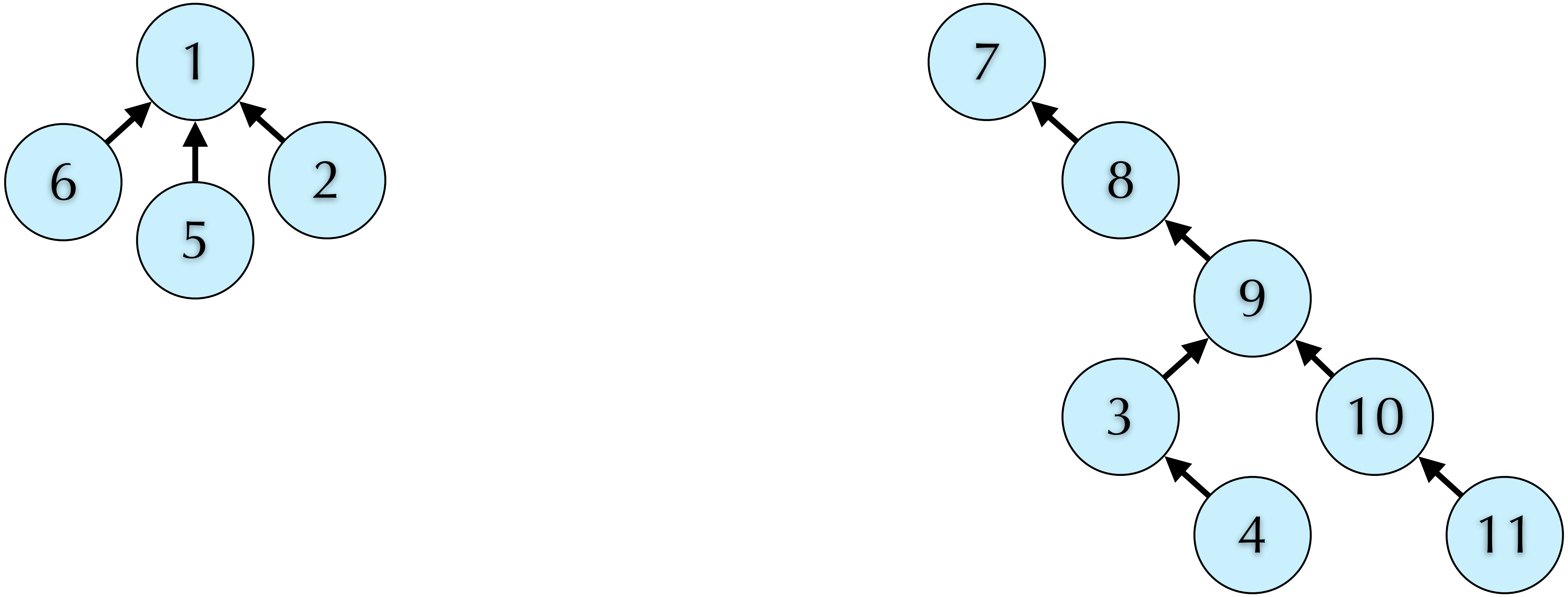
find(6)



Find w/ compression

i	1	2	3	4	5	6	7	8	9	10	11
r_i	1	1	7	7	1	1	7	7	7	7	7
q_i	1	1	9	3	1	5	7	7	8	9	10

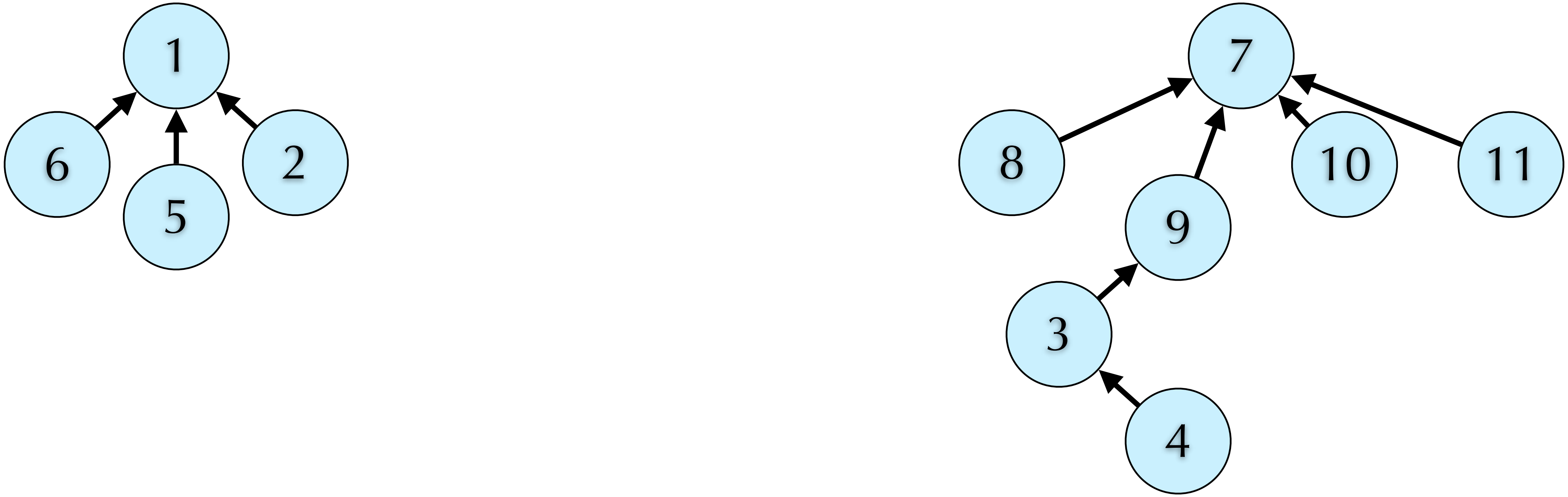
find(11)



Find w/ compression

i	1	2	3	4	5	6	7	8	9	10	11
r_i	1	1	7	7	1	1	7	7	7	7	7
q_i	1	1	9	3	1	5	7	7	7	7	7

find(11)



Homework 6.1

- a) Give a iterative implementation of find with compression.
- b) Give a recursive implementation of find with compression.
- c) Modify the implementation on the slides to support SetSize
- d) What is the time complexity of Find?
- e) Give an implementation which guarantees O(logn)-time Union, Find, SameSet and SetSize.