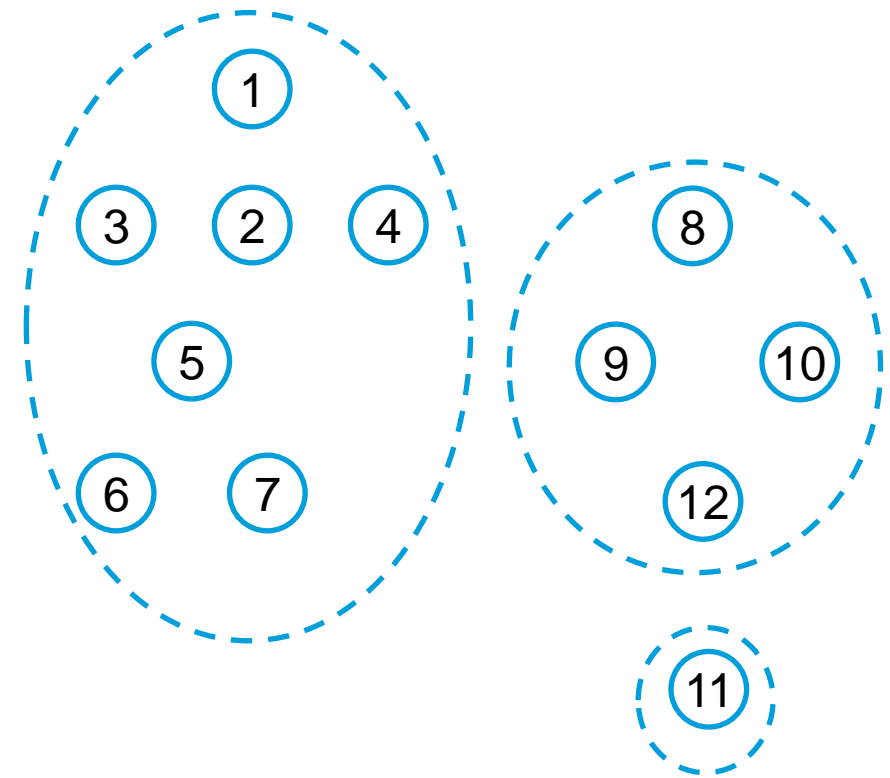


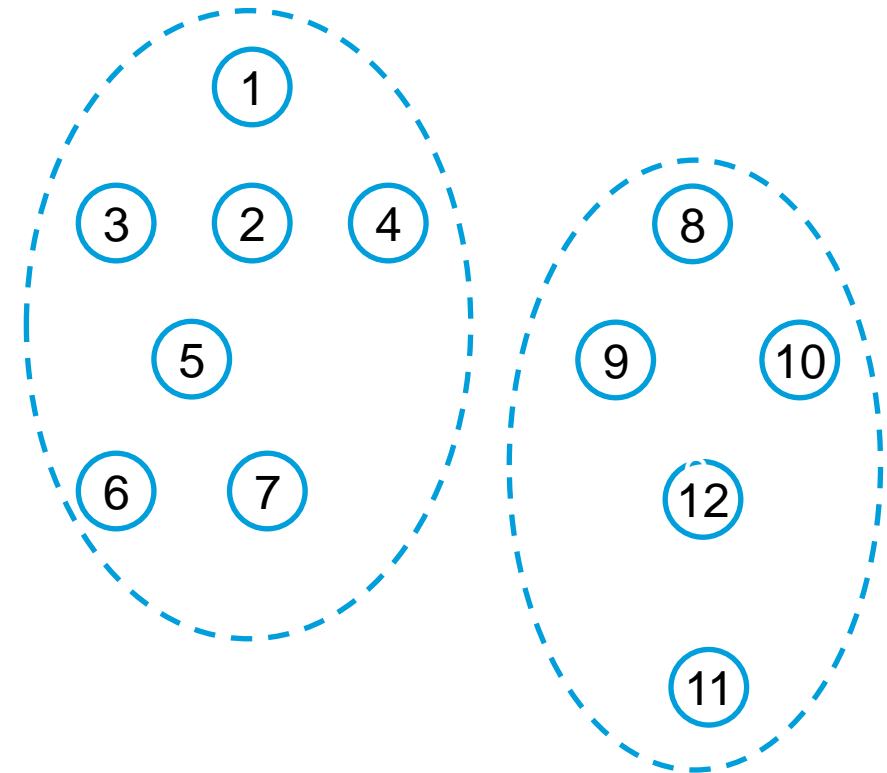
Cấu trúc Disjoint Set (Union - Find)

- Disjoint Set: Là cấu trúc dữ liệu biểu diễn các tập không giao nhau với 2 thao tác chính
 - Find(x): trả về định danh của tập chứa x
 - Unify(r1, r2): Hợp nhất 2 tập hợp định danh là r1 và r2 làm một



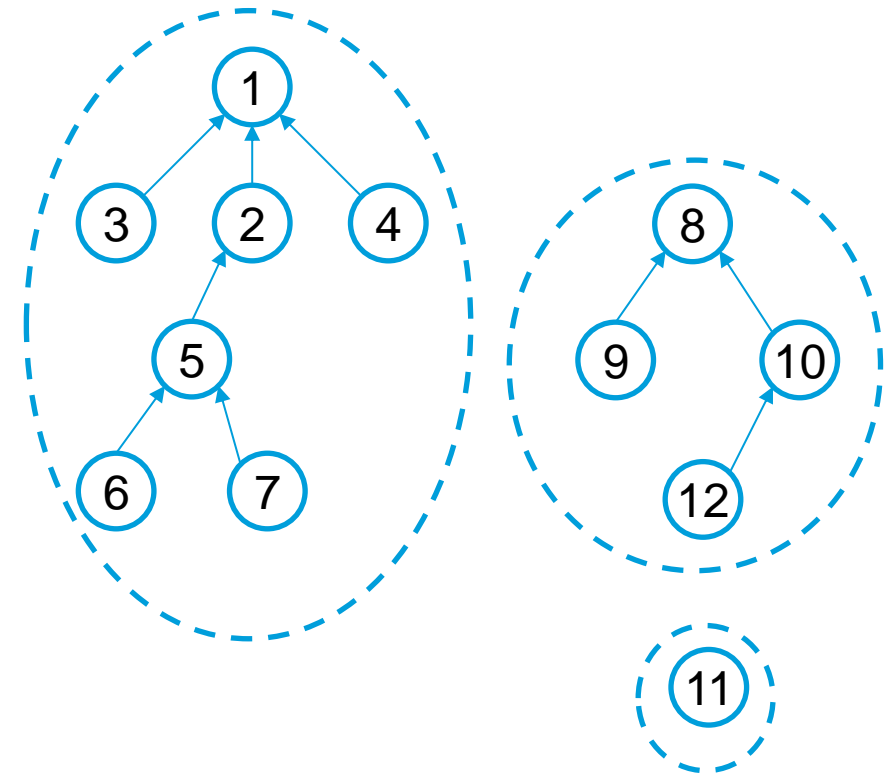
Cấu trúc Disjoint Set (Union - Find)

- Disjoint Set: Là cấu trúc dữ liệu biểu diễn các tập không giao nhau với 2 thao tác chính
 - Find(x): trả về định danh của tập chứa x
 - Unify(r1, r2): Hợp nhất 2 tập hợp định danh là r1 và r2 làm một



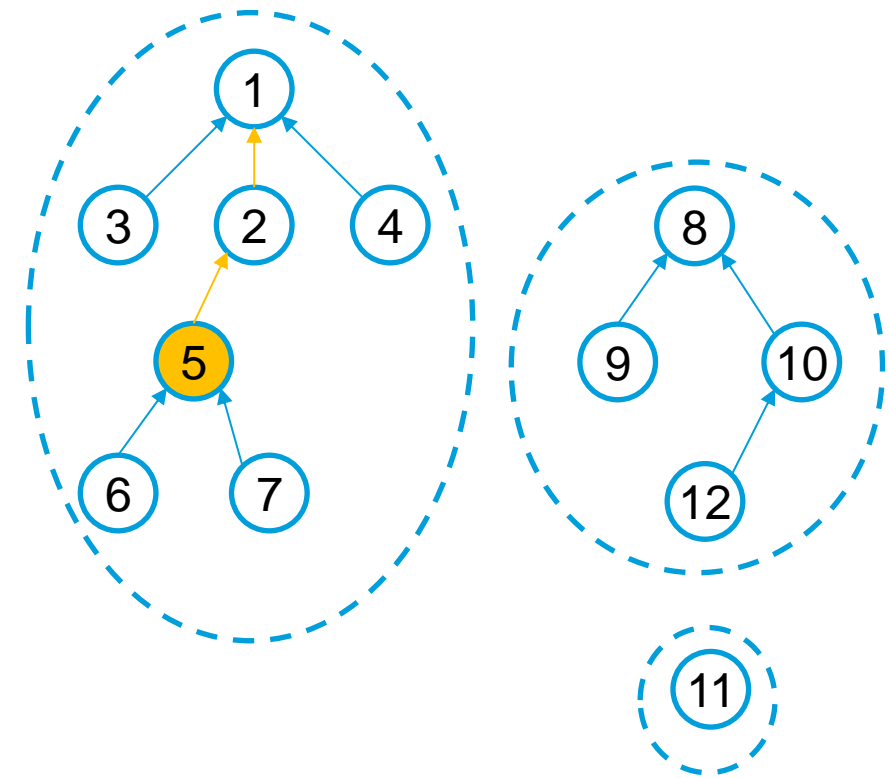
Cấu trúc Disjoint Set (Union - Find)

- Disjoint Set: Là cấu trúc dữ liệu biểu diễn các tập không giao nhau với 2 thao tác chính
 - Find(x): trả về định danh của tập chứa x
 - Unify(r1, r2): Hợp nhất 2 tập hợp định danh là r1 và r2 làm một
- Mỗi tập được biểu diễn bởi cây có gốc
 - Mỗi nút của cây là một phần tử
 - Mỗi nút có 1 nút cha duy nhất (cha của nút gốc là chính nó)
 - Nút gốc là định danh của tập



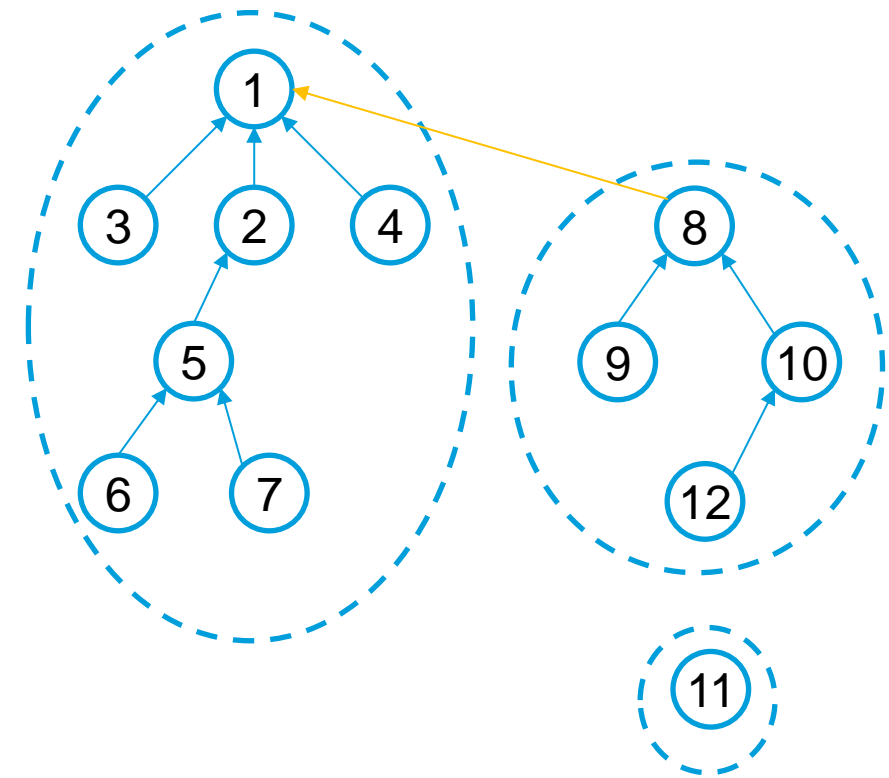
Cấu trúc Disjoint Set (Union - Find)

- Disjoint Set: Là cấu trúc dữ liệu biểu diễn các tập không giao nhau với 2 thao tác chính
 - Find(x): trả về định danh của tập chứa x
 - Unify(r1, r2): Hợp nhất 2 tập hợp định danh là r1 và r2 làm một
- Mỗi tập được biểu diễn bởi cây có gốc
 - Mỗi nút của cây là một phần tử
 - Mỗi nút có 1 nút cha duy nhất (cha của nút gốc là chính nó)
 - Nút gốc là định danh của tập



Cấu trúc Disjoint Set (Union - Find)

- Disjoint Set: Là cấu trúc dữ liệu biểu diễn các tập không giao nhau với 2 thao tác chính
 - Find(x): trả về định danh của tập chứa x
 - Unify(r1, r2): Hợp nhất 2 tập hợp định danh là r1 và r2 làm một
- Mỗi tập được biểu diễn bởi cây có gốc
 - Mỗi nút của cây là một phần tử
 - Mỗi nút có 1 nút cha duy nhất (cha của nút gốc là chính nó)
 - Nút gốc là định danh của tập



Cấu trúc Disjoint Set (Union - Find)

- Cài đặt
 - $p[x]$: là nút cha của x trên cây
 - $r[x]$: là hạng của nút x

```
makeSet(x){  
    p[x] = x; r[x] = 0;  
}  
Find(x){  
    if(x != p[x]) p[x] = Find(p[x]);  
    return p[x];  
}  
Unify(x, y){  
    if(r[x] > r[y]) p[y] = x;  
    else{  
        p[x] = y;  
        if(r[x] == r[y]) r[y] = r[y] + 1;  
    }  
}
```

