

Effectuer l'algorithme de Kruskal avec Union-Find.

On note les arêtes avec les 2 sommets séparés par un tiret, suivi du poids séparé par une virgule.

### Kruskal

#### Initialisation

Queue de priorité (début de la queue à gauche)

4-7, 1	1-6, 2	6-7, 3	2-4, 4	1-4, 5	1-2, 6	2-5, 7	2-3, 8	1-7, 9	3-6, 10	3-5, 11	4-5, 12
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	------------	------------	------------

Arrêtes de l'arbre recouvrant de poids min

--	--	--	--	--	--	--	--	--	--	--	--

#### Union-Find



#### Itération 1

Connected(4, 7) -> false

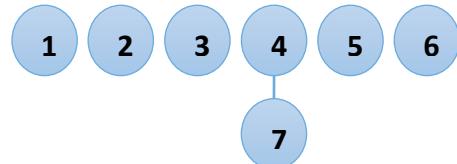
Union(4, 7)

Queue de priorité (début de la queue à gauche)

1-6, 2	6-7, 3	2-4, 4	1-4, 5	1-2, 6	2-5, 7	2-3, 8	1-7, 9	3-6, 10	3-5, 11	4-5, 12	
-----------	-----------	-----------	-----------	-----------	-----------	-----------	-----------	------------	------------	------------	--

### Arrêtes de l'arbre recouvrant de poids min

## Union-Find



## Itération 2

Connected(1, 6) -> false

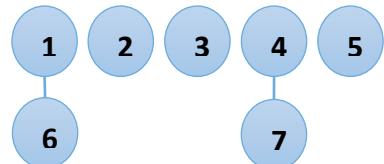
Union(1, 6)

### Queue de priorité (début de la queue à gauche)

6-7, 3	2-4, 4	1-4, 5	1-2, 6	2-5, 7	2-3, 8	1-7, 9	3-6, 10	3-5, 11	4-5, 12		
-----------	-----------	-----------	-----------	-----------	-----------	-----------	------------	------------	------------	--	--

### Arrêtes de l'arbre recouvrant de poids min

## Union-Find



## Itération 3

Connected(6, 7) -> false

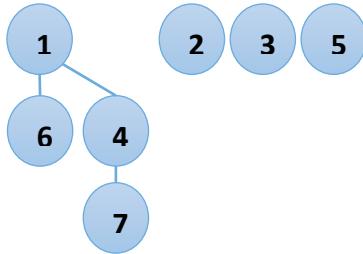
Union(6, 7)

## Queue de priorité (début de la queue à gauche)

2-4, 4	1-4, 5	1-2, 6	2-5, 7	2-3, 8	1-7, 9	3-6, 10	3-5, 11	4-5, 12		
-----------	-----------	-----------	-----------	-----------	-----------	------------	------------	------------	--	--

### Arrêtes de l'arbre recouvrant de poids min

### Union-Find



### Itération 4

Connected(2, 4) -> false

Union(2, 4)

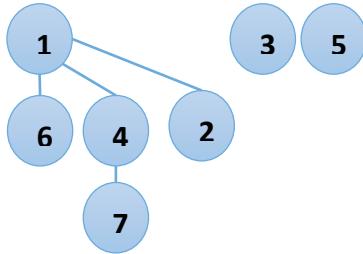
Queue de priorité (début de la queue à gauche)

1-4, 5	1-2, 6	2-5, 7	2-3, 8	1-7, 9	3-6, 10	3-5, 11	4-5, 12				
-----------	-----------	-----------	-----------	-----------	------------	------------	------------	--	--	--	--

Arrêtes de l'arbre recouvrant de poids min

4-7, 1	1-6, 2	6-7, 3	2-4, 4								
-----------	-----------	-----------	-----------	--	--	--	--	--	--	--	--

### Union-Find



### Itération 5

Connected(1, 4) -> true

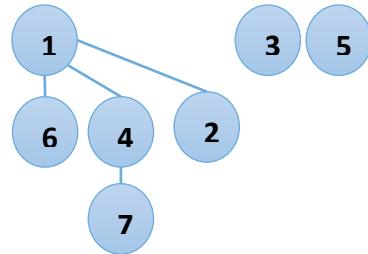
Queue de priorité (début de la queue à gauche)

1-2, 6	2-5, 7	2-3, 8	1-7, 9	3-6, 10	3-5, 11	4-5, 12					
-----------	-----------	-----------	-----------	------------	------------	------------	--	--	--	--	--

Arrêtes de l'arbre recouvrant de poids min

4-7, 1	1-6, 2	6-7, 3	2-4, 4								
-----------	-----------	-----------	-----------	--	--	--	--	--	--	--	--

### Union-Find



### Itération 6

Connected(1, 2) -> true

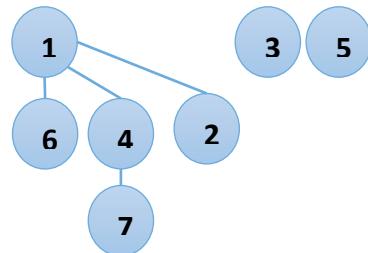
Queue de priorité (début de la queue à gauche)

2-5, 7	2-3, 8	1-7, 9	3-6, 10	3-5, 11	4-5, 12						
-----------	-----------	-----------	------------	------------	------------	--	--	--	--	--	--

Arrêtes de l'arbre recouvrant de poids min

4-7, 1	1-6, 2	6-7, 3	2-4, 4								
-----------	-----------	-----------	-----------	--	--	--	--	--	--	--	--

### Union-Find



### Itération 7

Connected(2, 5) -> false

Union(2, 5)

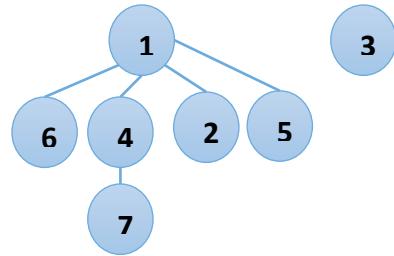
Queue de priorité (début de la queue à gauche)

2-3, 8	1-7, 9	3-6, 10	3-5, 11	4-5, 12							
-----------	-----------	------------	------------	------------	--	--	--	--	--	--	--

Arrêtes de l'arbre recouvrant de poids min

4-7, 1	1-6, 2	6-7, 3	2-4, 4	2-5, 7							
-----------	-----------	-----------	-----------	-----------	--	--	--	--	--	--	--

### Union-Find



### Itération 8

Connected(2, 3) -> false

Union(2, 3)

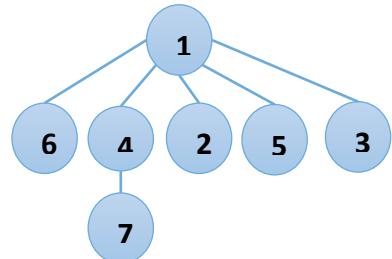
Queue de priorité (début de la queue à gauche)

1-7, 9	3-6, 10	3-5, 11	4-5, 12							
-----------	------------	------------	------------	--	--	--	--	--	--	--

Arrêtes de l'arbre recouvrant de poids min

4-7, 1	1-6, 2	6-7, 3	2-4, 4	2-5, 7	2-3, 8					
-----------	-----------	-----------	-----------	-----------	-----------	--	--	--	--	--

### Union-Find



On arrête car on a 6 arrêtes dans le MST.

Arbre recouvrant de poids minimum final (arrêtes choisies en rouge)

