Init: edges: [(1,4,3), (0,2,5),
(0,3,6), (2,3,6), (1,2,7),
(3,4,8), (1,3,11)]
parents: [0 1 2 3 4]
ranks: [0 0 0 0 0]

Iter 1: edge (1,4,3)   ~~rootof~~ rootof 1 = 1, rootof 4 = 4

parents [0 1 2 3 1]
ranks: [0 1 0 0 0]

①—3—④
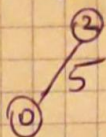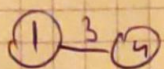
Iter 2: edge (0,2,5)   rootof 0 = 0, rootof 2 = 2

parents: [0 1 0 3 1]
ranks: [1 1 0 0 0]

①—3—④

②
5
⓪

Iter 3: edge (0,3,6)   rootof 0 = 0, rootof 3 = 3

parents: [0 1 0 0 1]
ranks: [1 1 0 0 0]

①—3—④
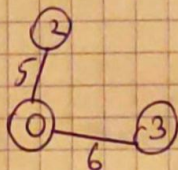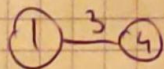
②
5
⓪—6—③

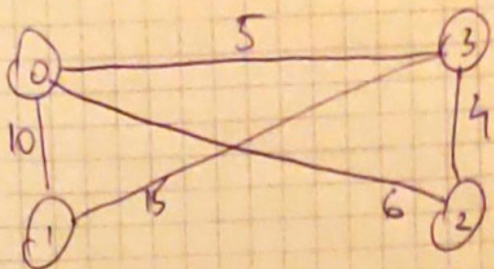Iter 4: edge (2,3,6)   rootof 2 = 0, rootof 3 = 0
roots are equal ⇒ edge would cause
a cycle.

Iter 5: edge (1, 2, 7) . rootof 1 = 1   rootof 2 = 0
parents: [1 1 0 0 1]
ranks : [1 2 0 0 0]
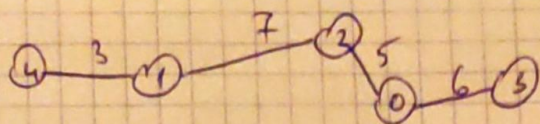
The algorithm stops because we have V-1
edges in our result.

Manual exec - Kruskal

Init: edges $[(2,3,4),(0,3,5),(0,2,6),(0,1,10),(1,2,15)]$
parents $[0 \ 1 \ 2 \ 3]$ ranks $[0 \ 0 \ 0 \ 0]$

Iter 1: edge = $(2,3,4)$ rootof 2 = 2, rootof 3 = 3
rootof 2 != rootof 3 => no cycles
parents $[0 \ 1 \ 2 \ 2]$ ranks $[0 \ 0 \ 1 \ 0]$

$(2) \overset{4}{—} (3)$

Iter 2: edge = $(0,3,5)$ rootof 0 = 0, rootof 3 = 2
rootof 0 != rootof 3 => no cycles
parents $[2 \ 1 \ 2 \ 2]$ ranks $[0 \ 0 \ 1 \ 0]$

$(2) \overset{4}{—} (3) \overset{5}{—} (0)$

Iter 3: edge = $(0,2,6)$ rootof 2 = 2, rootof 0 = 2
rootof 0 == rootof 2 => this edge would cause cycles
skip.

Iter 4: edge = $(0,1,10)$ rootof 0 = 2, rootof 1 = 1
rootof 0 != rootof 1
parents $[2 \ 2 \ 2 \ 2]$ ranks $[0 \ 0 \ 1 \ 0]$

$(1) \overset{10}{—} (0) \overset{5}{—} (3) \overset{4}{—} (2)$

The algorithm stops because we arrived at len(result)
equal to the nr. of vertices - 1.