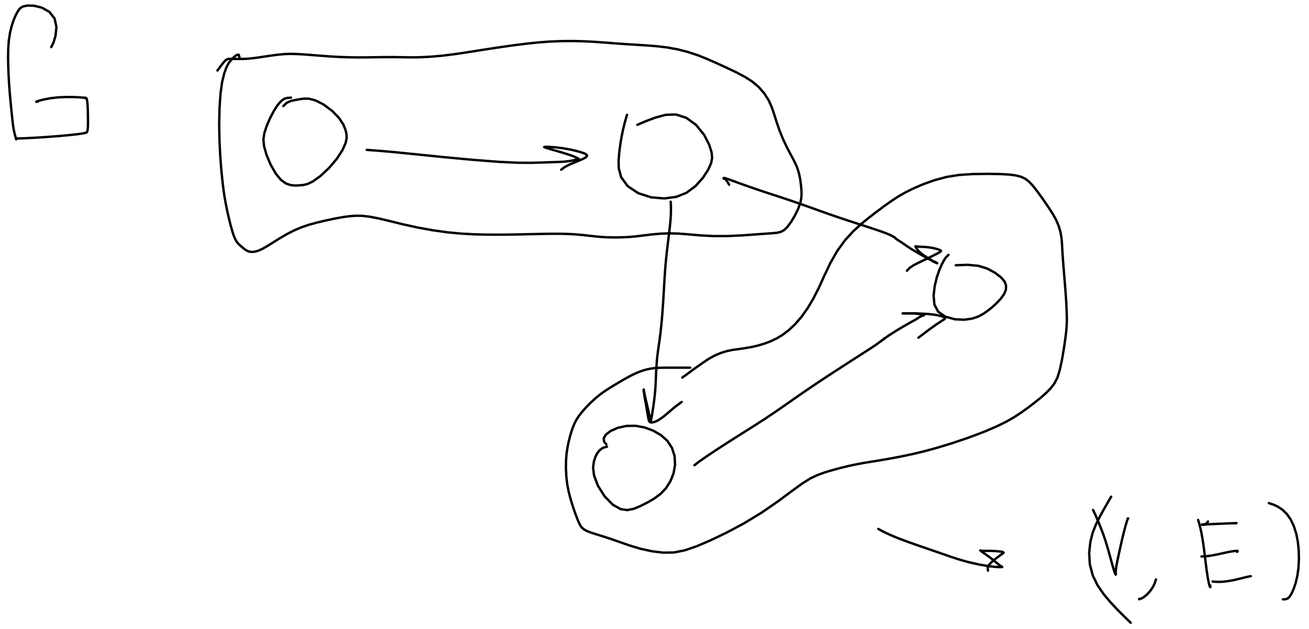
Algorithms & Data Structures III: Boruvka's Algorithm

Mikhail Anukhin


G
(V, E)

Algorithm Description

1) Init. trivial tree T on n nodes.
2) Divide G in n components.
3) Find minimal adjacency edge for every component and add to tree T
4) Repeat Step 3 until there is only one component

Pseudocode

```
function boruvkaMST():
  while T.size < n - 1:
    for k in Component:
      w(minEdge[k]) = inf
    InitializeCompents(T)
    for (u, v) in E:
      if u.comp != v.comp
        if w(minEdge[u.comp]) > w(u, v):
          minEdge[u.comp] = (u, v)
        if w(minEdge[v.comp]) > w(u, v):
          minEdge[v.comp] = (u, v)
    for k in Component:
      T.addEdge(minEdge[k])
  return T
```

log n times

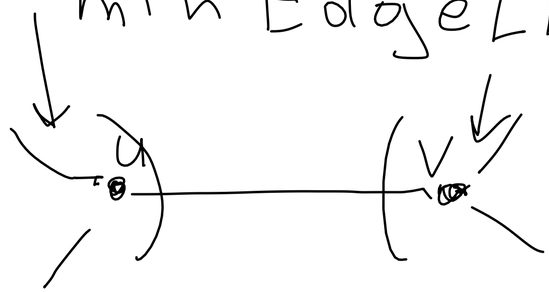
E

Time Complexity: E.LogV

T

Component-set of con.comp. in T

minEdge[1...n]



i-th

k comp

(i-1)-th

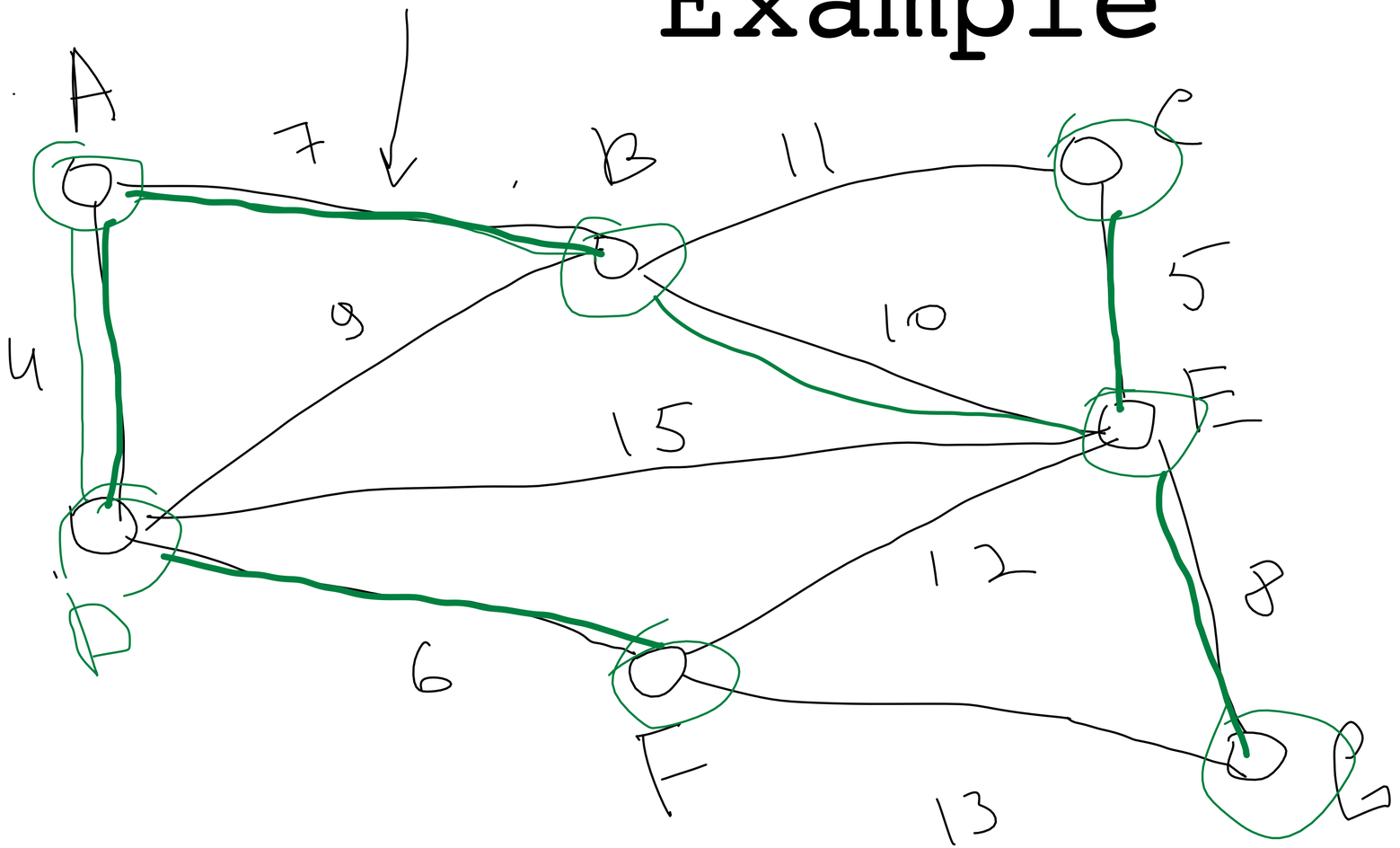
2K

k

(i-2)-th

4K

Example



0 1 2 3

A B D F

C E G

A B D F

10

C E G

10

Prim's

Kruskal's

Boruvka's