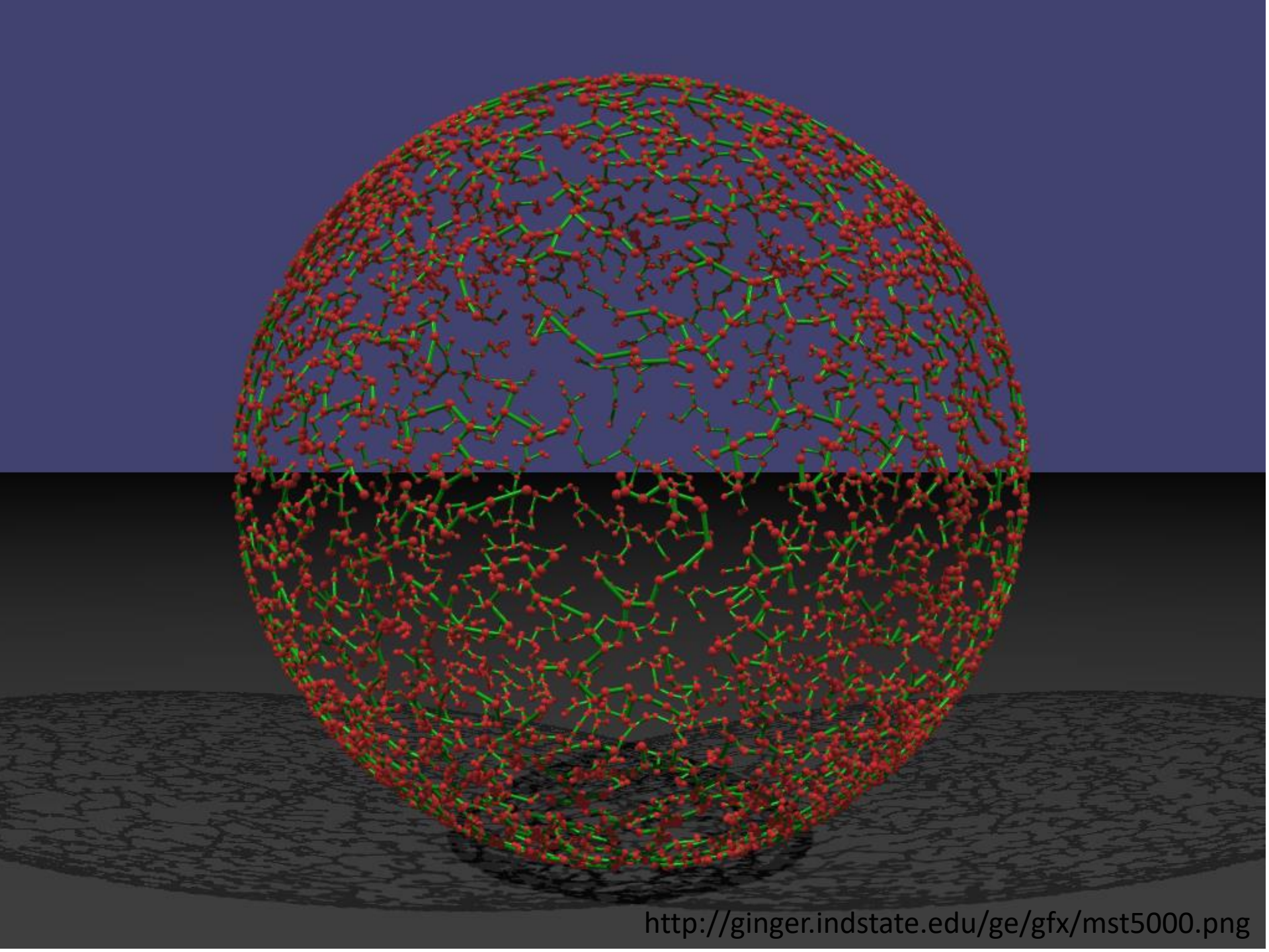
Minimális feszítőfa

9. ELŐADÁS

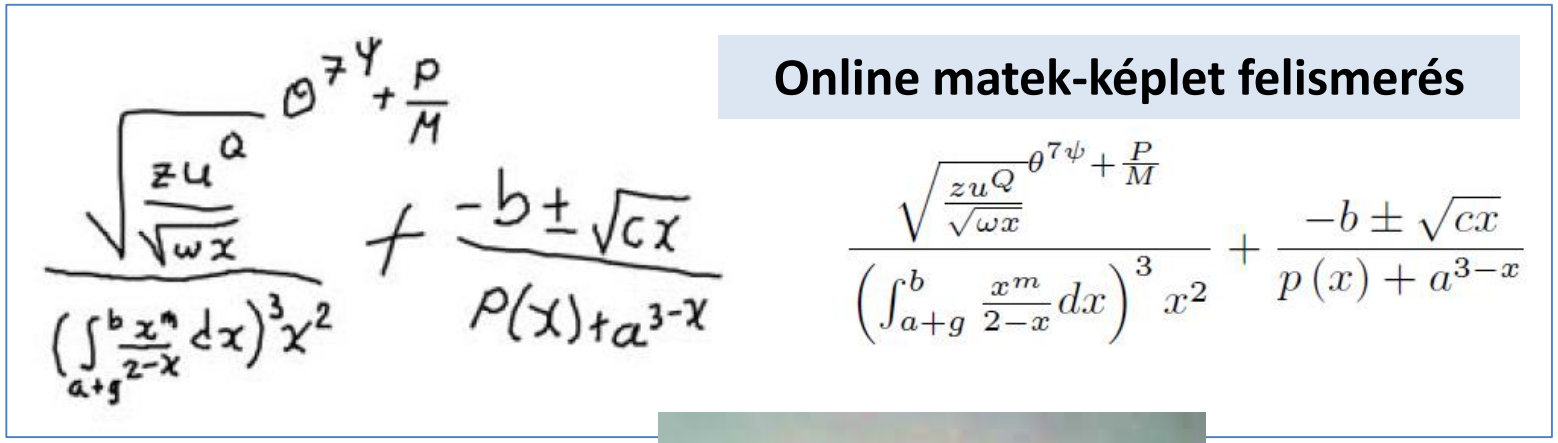
Sapientia-EMTE

2020-21

http://algs4.cs.princeton.edu/lectures/43MinimumSpanningTrees.pdf



http://ginger.indstate.edu/ge/gfx/mst5000.png



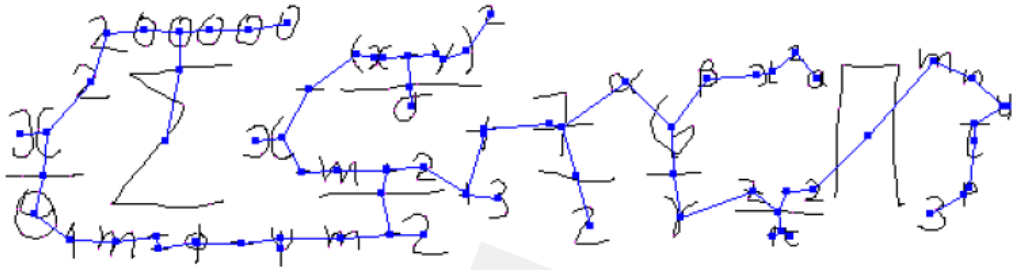
Valós idejű arcfelismerés

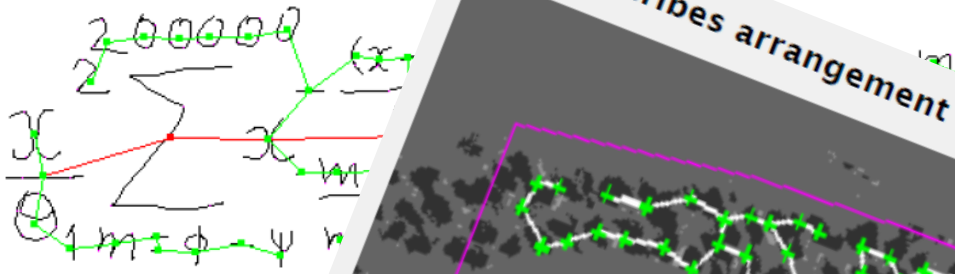
CANCER RESEARCH UK

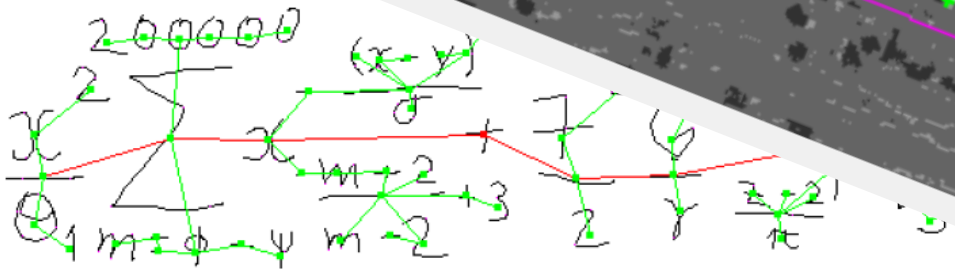
Ökotoxikológia

Minimális feszítőfa

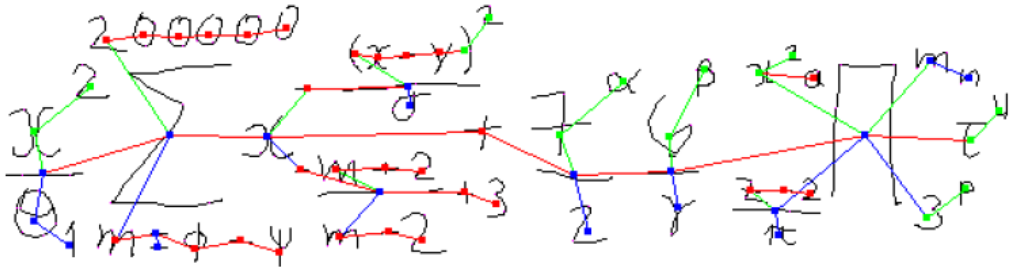
- "Recognition of On-line Handwritten Mathematical Expressions Using a **Minimum Spanning Tree** Construction and Symbol Dominance"
- "Heuristic potency of the **minimum spanning tree** (MST) method in toxicology".
- "Real-Time Salient Object Detection with a **Minimum Spanning Tree**"
- "Efficient regionalization techniques for socio-economic geographical units using **minimum spanning trees**"
- "A **Minimum Spanning Forest** Based Method for Noninvasive Cancer Detection with Hyperspectral Imaging"



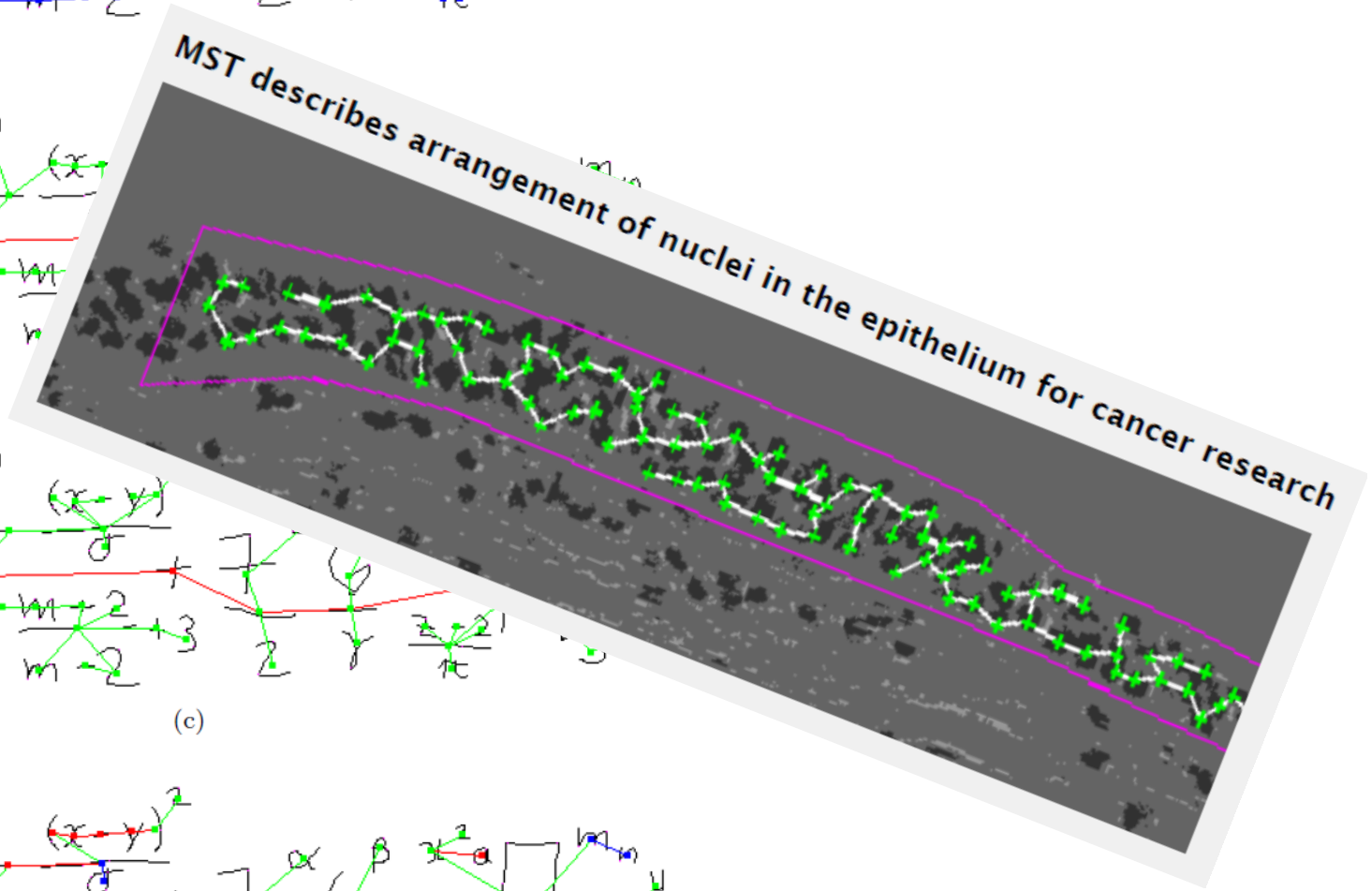




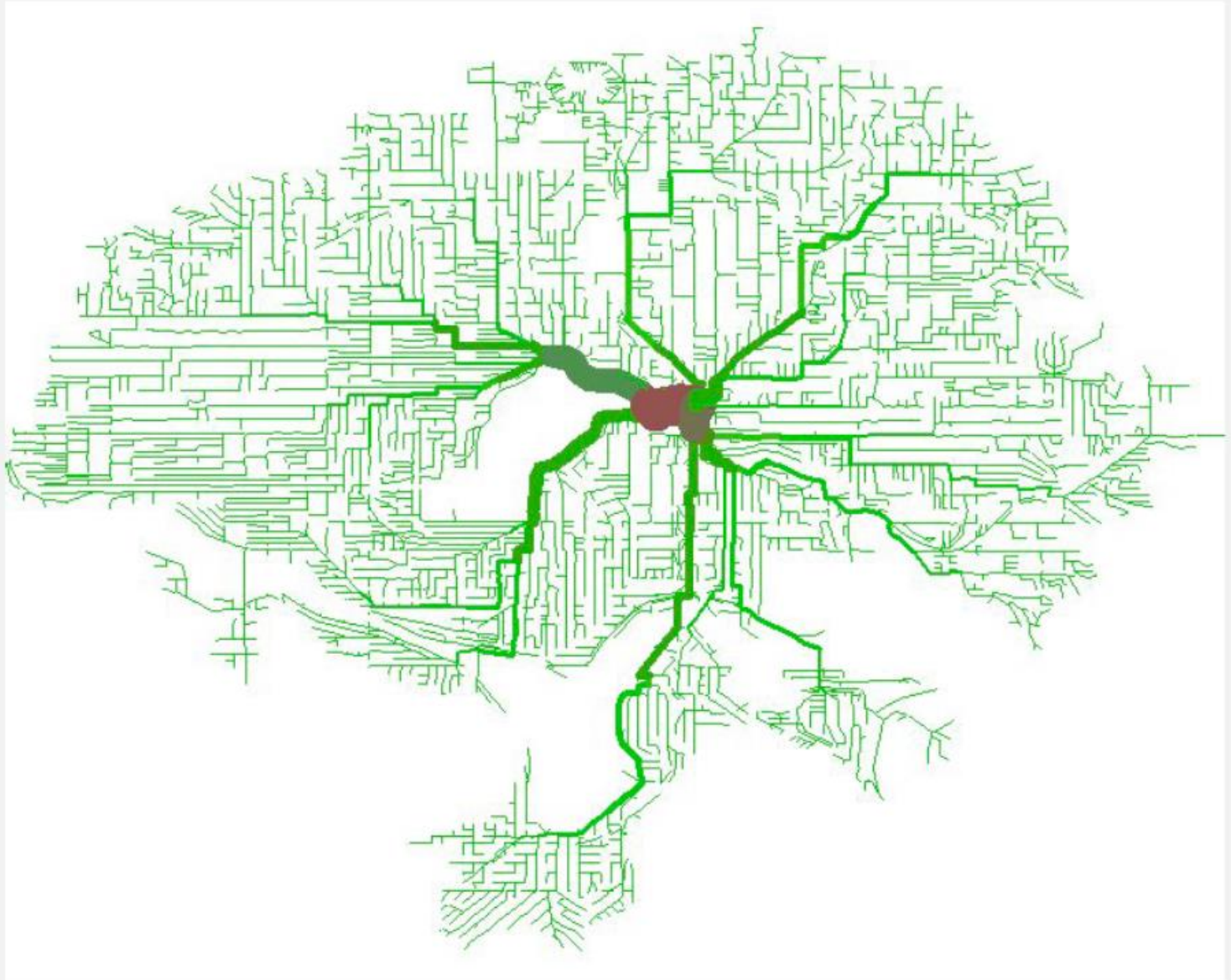
(c)



(d)


MST describes arrangement of nuclei in the epithelium for cancer research

MST of bicycle routes in North Seattle



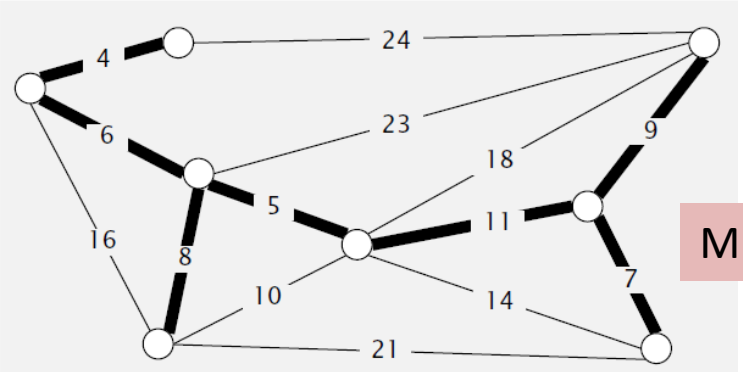
http://www.flickr.com/photos/ewedistrict/21980840

Otakar Borůvka

- The problem of designing efficient electric distribution networks had been suggested to Borůvka by his friend Jindřich Saxel, an employee of the West Moravian Power Company, during World War I. In his **1926** paper *O jistém problému minimálním* (English *On a certain minimal problem*), Borůvka solved this problem by modeling it mathematically as a **minimum spanning tree** problem, and described **the first known algorithm** for finding the minimum spanning tree of a metric space (the set of cities to be connected by the network, together with their distances). (*wikipedia*)

Minimális **költségű** feszítőfa

- **Feszítő részgráf** *(spanning subgraph)*
  - élek törlése révén
- **Feszített részgráf** *(induced subgraph)*
  - pontok törlése révén
- Egy gráfnak több feszítőfája lehet
- Súlyozott gráfokban:
  - **minimális költségű feszítőfa** (MST: minimum spanning tree)



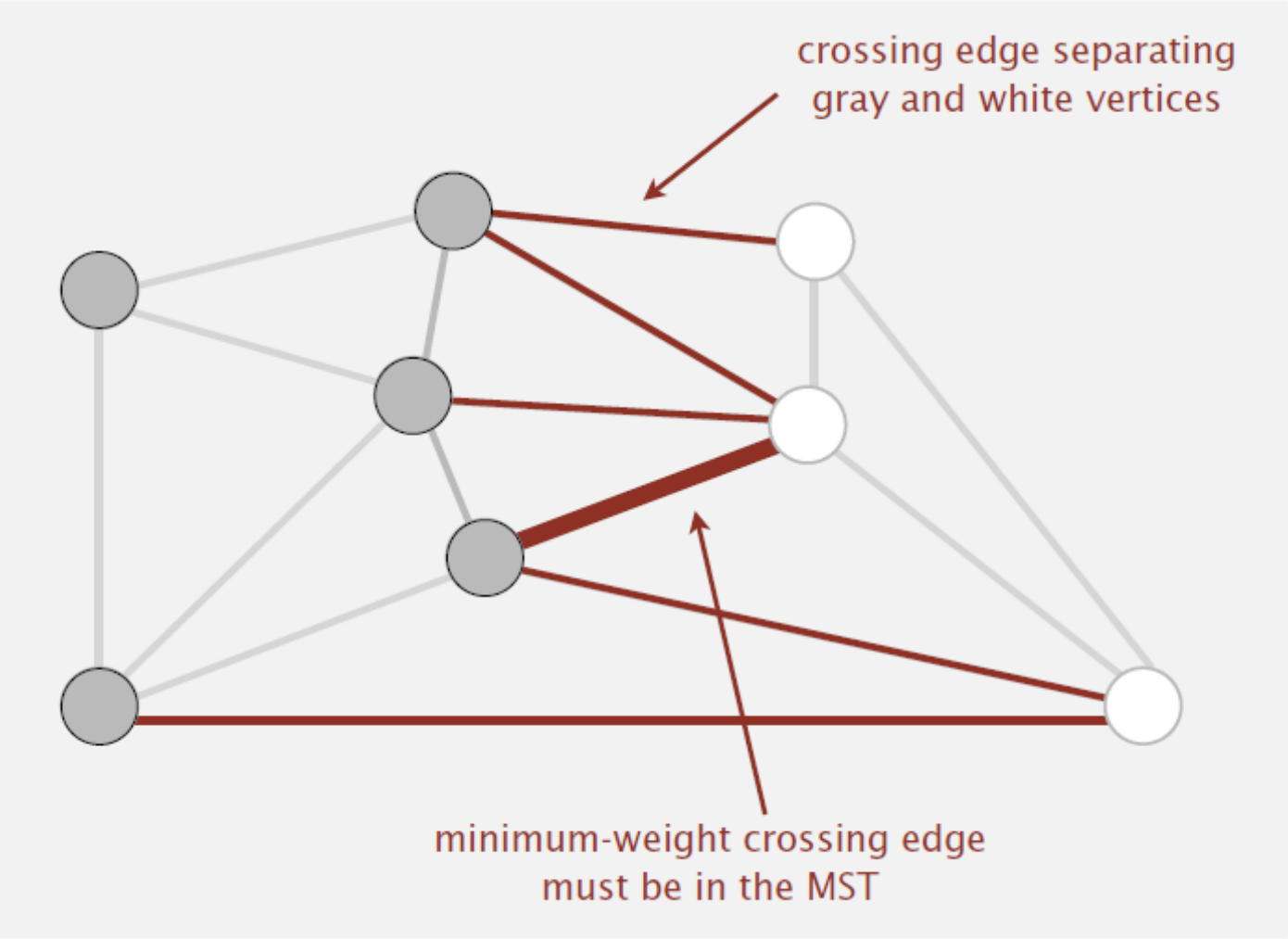
minimum spanning tree T
(cost = 50 = 4 + 6 + 8 + 5 + 11 + 9 + 7)

Mi lenne a „nyers erő” módszere?

Tulajdonságai (1)

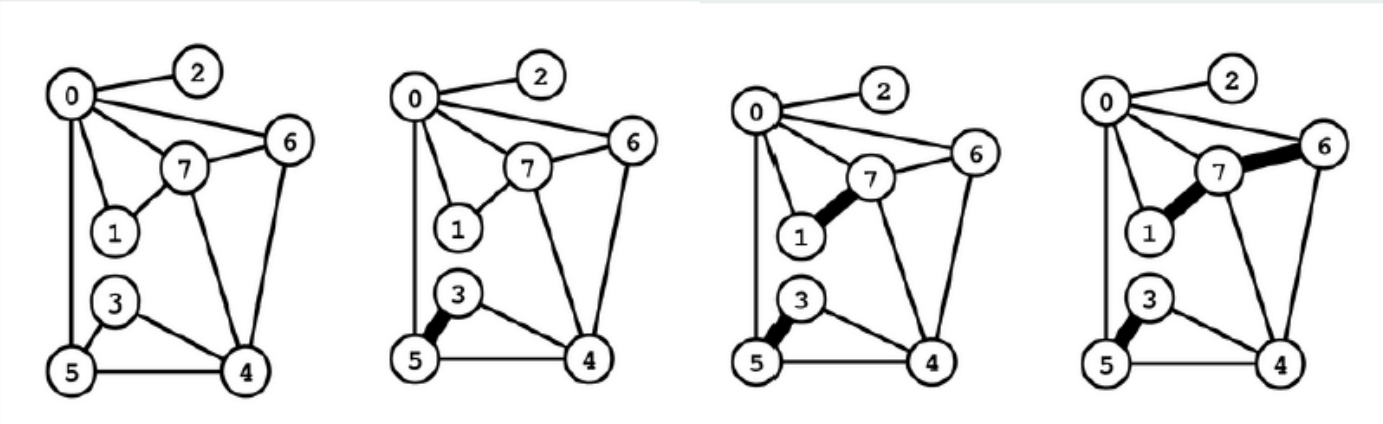
- Ha egy gráfban minden él költsége különböző, akkor a feszítő fa **egyértelmű**.
- Ha *C* **kör** a gráfban, akkor a **legnagyobb költségű éle** nem tartozik egy minimális költségű feszítő fához.
- Duálisan, ha *C* **vágás**, akkor a **legkisebb költségű éle** az összes minimális költségű feszítő fának.
- Ha a gráfban van egy egyértelmű legkisebb költségű él, akkor a gráf minimális költségű feszítő fáinak mindegyike tartalmazni fogja ezt az élt.

Tulajdonságai (2)

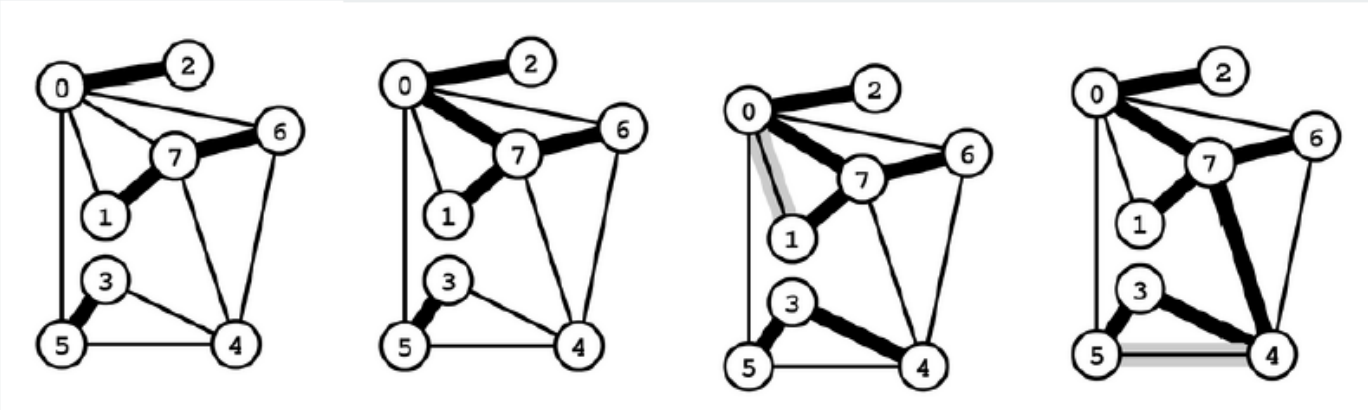


Kruskal's Algorithm: Example

Kruskal's algorithm. [Kruskal, 1956] Consider edges in ascending order of cost. Add the next edge to T unless doing so would create a cycle.



3-5 1-7 6-7

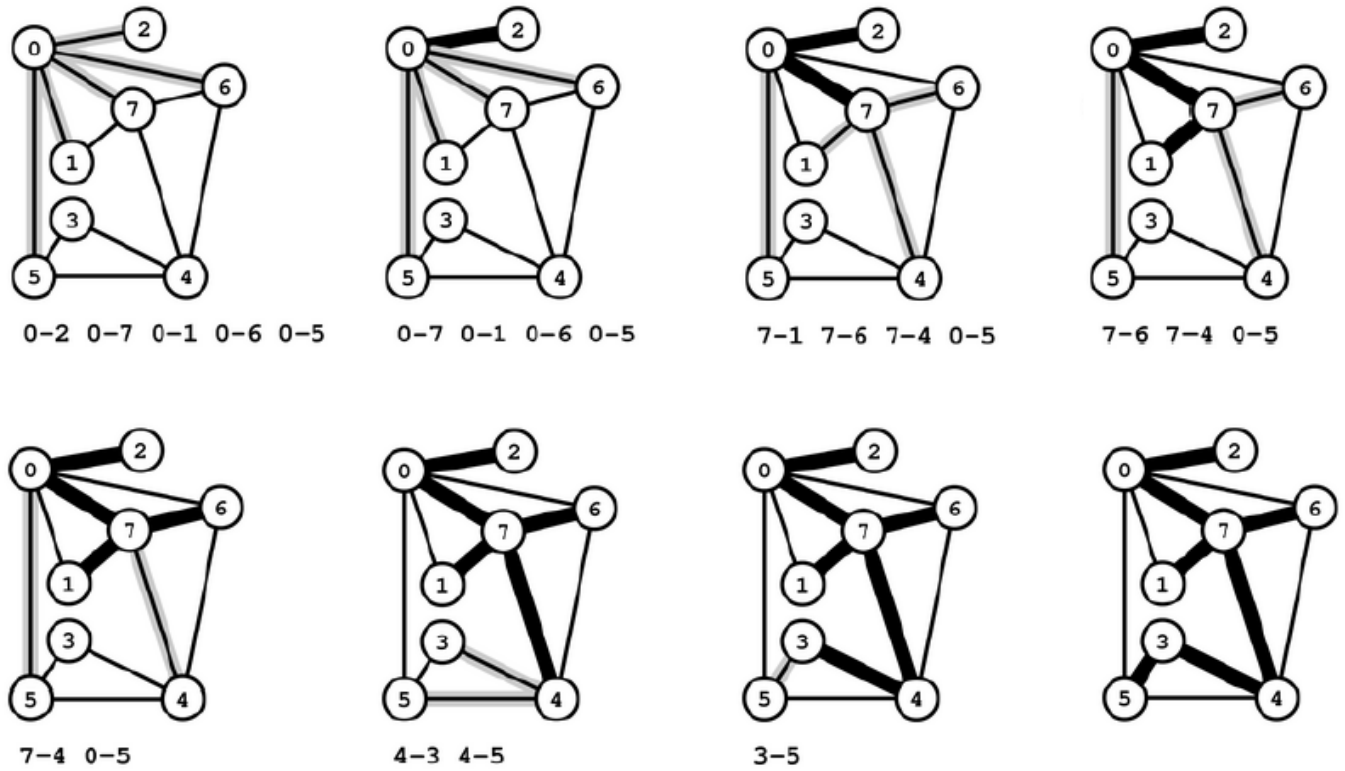


0-2 0-7 0-1 3-4 4-5 4-7

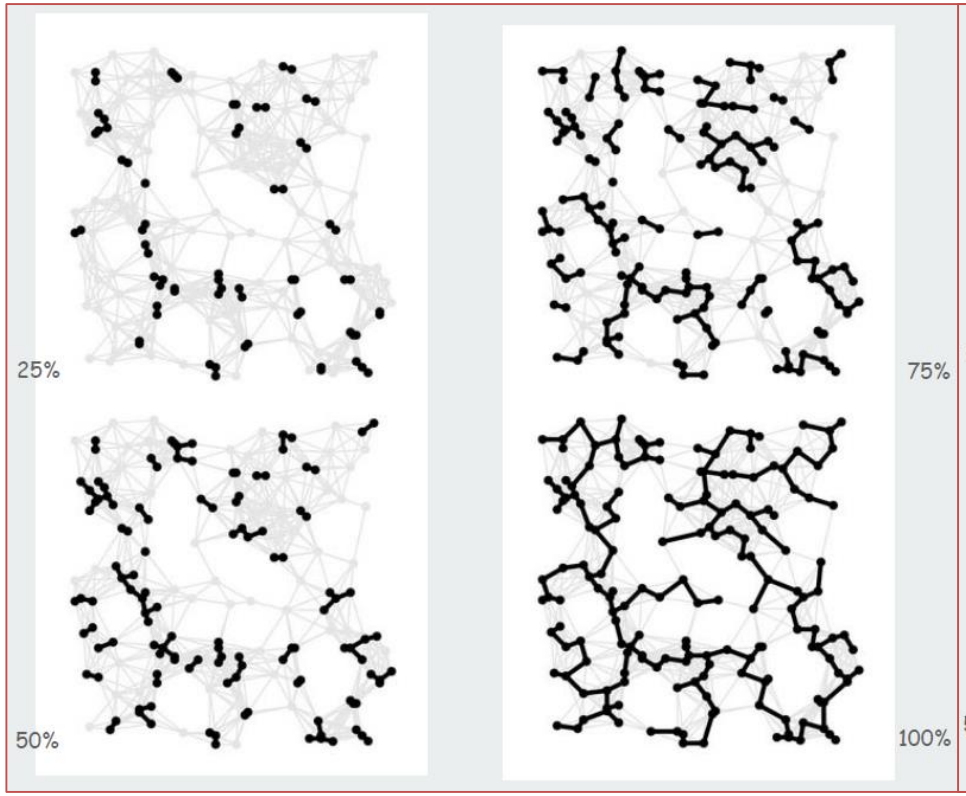
3-5	0.18
1-7	0.21
6-7	0.25
0-2	0.29
0-7	0.31
0-1	0.32
3-4	0.34
4-5	0.40
4-7	0.46
0-6	0.51
4-6	0.51
0-5	0.60

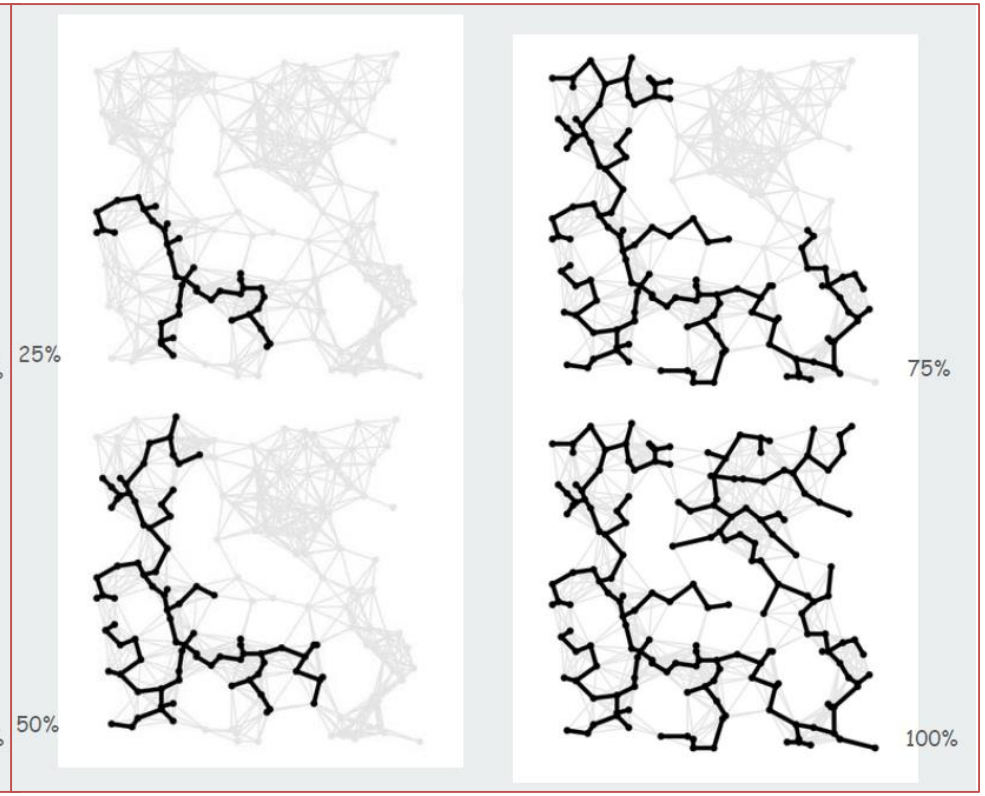
Prim's algorithm example

Prim's algorithm. [Jarník 1930, Dijkstra 1957, Prim 1959]
Start with vertex 0 and greedily grow tree T. At each step, add cheapest edge that has exactly one endpoint in T.

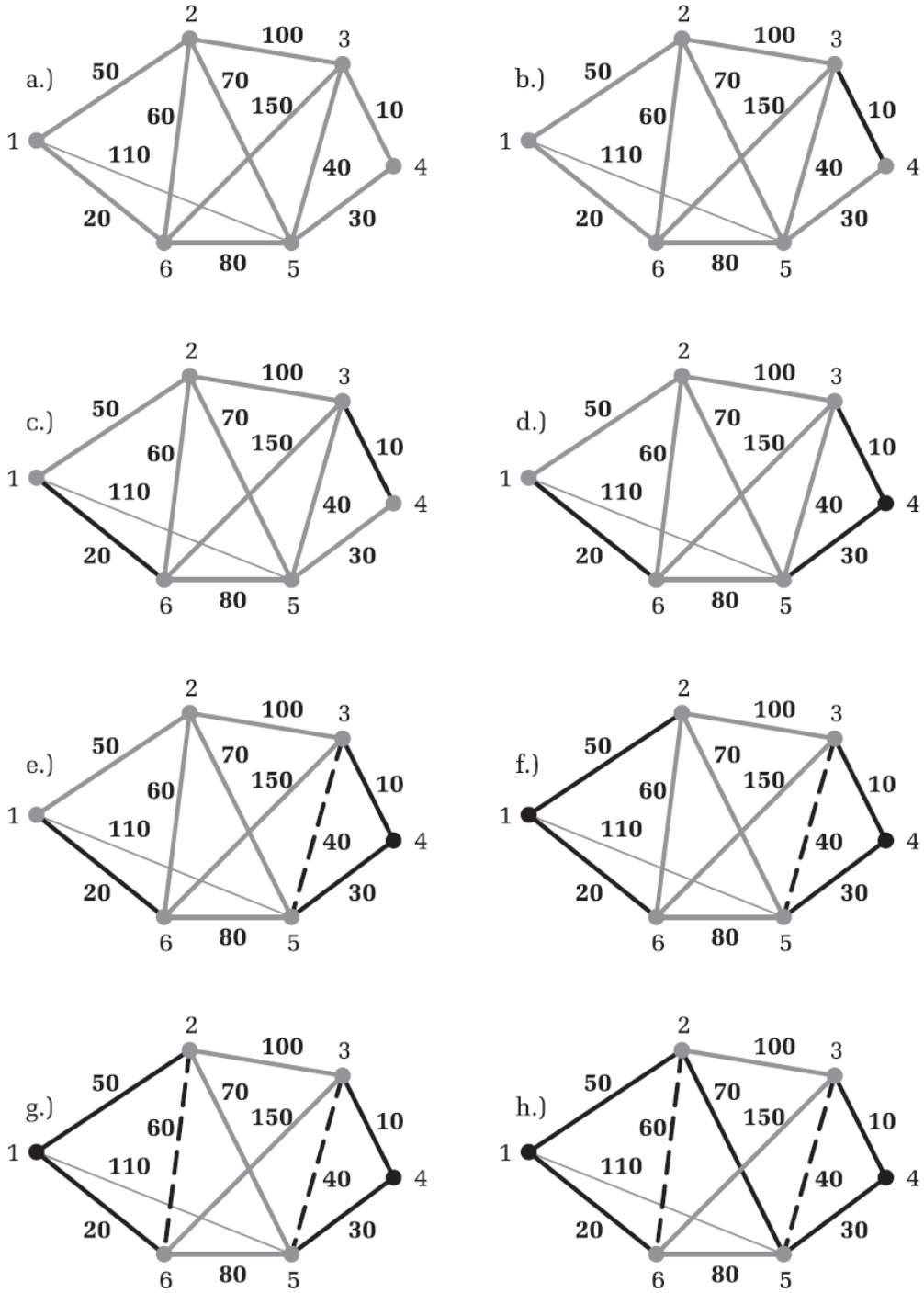


0-1	0.32
0-2	0.29
0-5	0.60
0-6	0.51
0-7	0.31
1-7	0.21
3-4	0.34
3-5	0.18
4-5	0.40
4-6	0.51
4-7	0.46
6-7	0.25


25%
75%
50%
100%


25%
75%
50%
100%

Kruskal algoritmusa



1	**2**	**3**	**4**	**5**	**6**	→	**1**	**2**	**3**	**3**	**5**	**6**
1	2	3	4	5	6		1	2	3	4	5	6

1	**2**	**3**	**3**	**5**	**6**	→	**1**	**2**	**3**	**3**	**5**	**1**
1	2	3	4	5	6		1	2	3	4	5	6

1	**2**	**3**	**3**	**5**	**1**	→	**1**	**2**	**3**	**3**	**3**	**1**
1	2	3	4	5	6		1	2	3	4	5	6

1	**2**	**3**	**3**	**3**	**1**
1	2	3	4	5	6

1	**2**	**3**	**3**	**3**	**1**	→	**1**	**1**	**3**	**3**	**3**	**1**
1	2	3	4	5	6		1	2	3	4	5	6

1	**1**	**3**	**3**	**3**	**1**
1	2	3	4	5	6

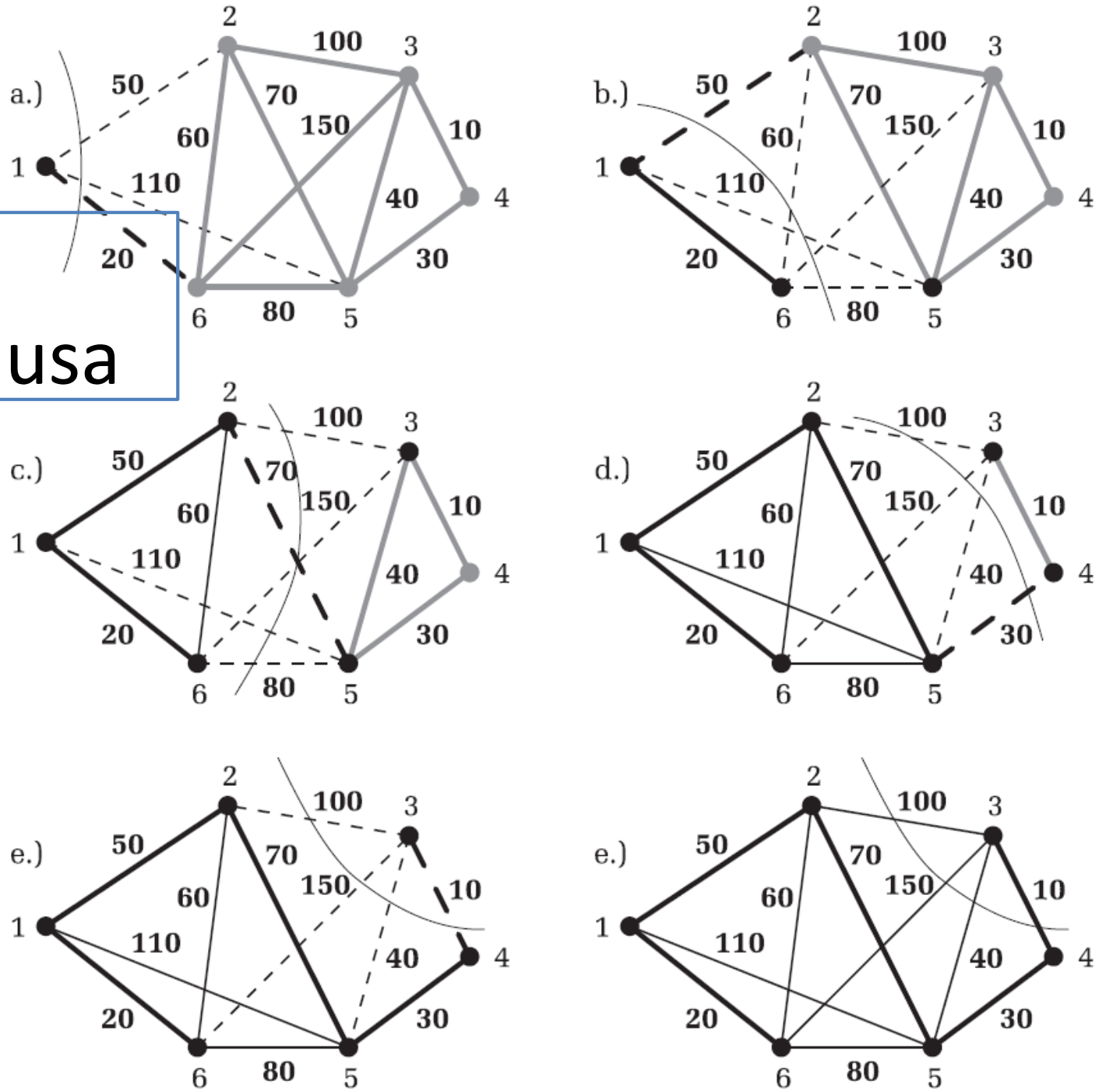
1	**1**	**3**	**3**	**3**	**1**	→	**1**	**1**	**1**	**1**	**1**	**1**
1	2	3	4	5	6		1	2	3	4	5	6

Kruskal algoritmusa

```
eljárás KRUSKAL(G)
   minden i ← 1,n végezd
      fa[i] ← i
   vége minden
   RENDEZÉS_SZÚLY_SZERINT(élek,m)
   össz_súly ← 0
   minden i ← 1,m végezd
      egyik = fa[élek[i].u]
      másik = fa[élek[i].v]
      ha egyik ≠ másik akkor
         össz_súly ← össz_súly + élek[i].súly
         kiír: élek[i].u, élek[i].v, élek[i].súly
         EGYESÍT(fa,n,egyik,másik)
      vége ha
   vége minden
   kiír: össz_súly
vége KRUSKAL
```

```
eljárás EGYESÍT(fa,n,egyik,másik)
   minden i ← 1,n végezd
      ha fa[i] = másik akkor
         fa[i] ← egyik
      vége ha
   vége minden
vége EGYESÍT
```

Prim algoritmusa



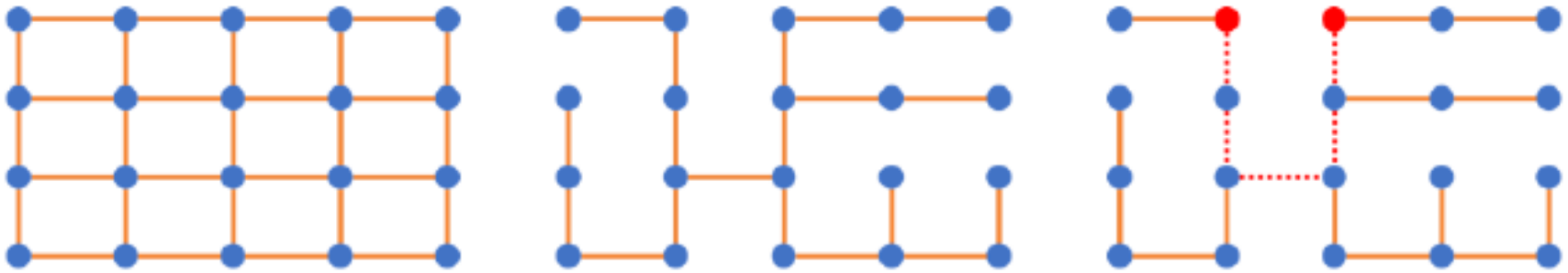
Prim algoritmusa

```
függvény PRIM(G,r)
   Q ← V(G)
   minden v ∈ Q végezd
      távolság[v] ← ∞
   vége minden
   távolság[r] ← 0
   apa[r] ← 0
   amíg Q ≠ ∅ végezd
      u ← KIVESZ_MIN(Q)
      minden v ∈ szomszéd(u) végezd
         ha (v ∈ Q ÉS (súly(u,v) < távolság[v])) akkor
            apa[v] ← u
            távolság[v] ← súly(u,v)
         vége ha
      vége minden
   vége amíg
   vissza apa
vége PRIM
```

KIVESZ_MIN(Q) – meghatározza a távolság tömb alapján a fához „legközelebb” lévő Q-beli csúcsot (azt a csúcsot, amelyet az aktuális *vágás* legkisebb súlyú éle köt a fához), és a fához kapcsolja (törli Q-ból). Az apa-tömbben regisztráljuk, hogy a fának melyik csúcsához csatoltuk az új csúcsot.

Viszonylag homogén háttérszínű képen jól elhatárolt objektum beazonosítása

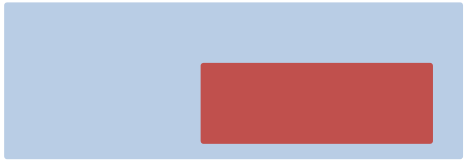
- Egy kép minimális feszítőfájának meghatározása
  - KÉP, mint irányítatlan gráf
    - Pontok: a pixelek
    - Élek: szomszédos pixelek közt (4 irányba)
    - Élek súlya: a végpontok szín-különbsége
  - MST: súlyuk szerinti csökkenő sorrendben töröljük az éleket, míg fához nem jutunk (Kruskal)
    - A nagy színkülönbségű szomszédok fabeli távolsága nagy lesz



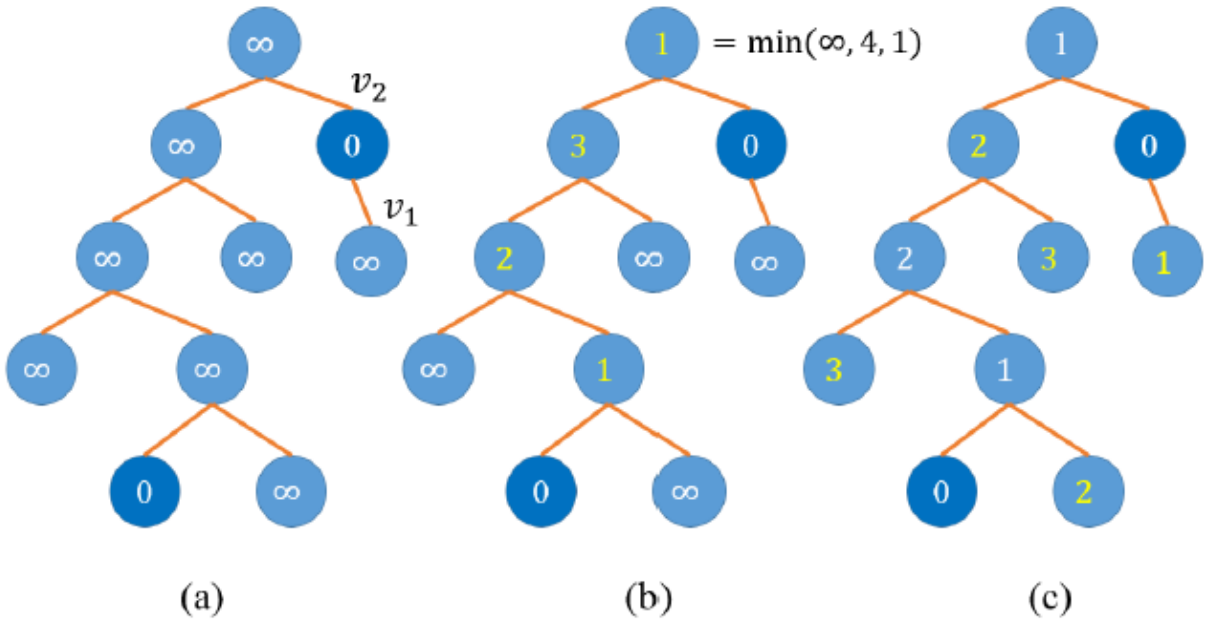
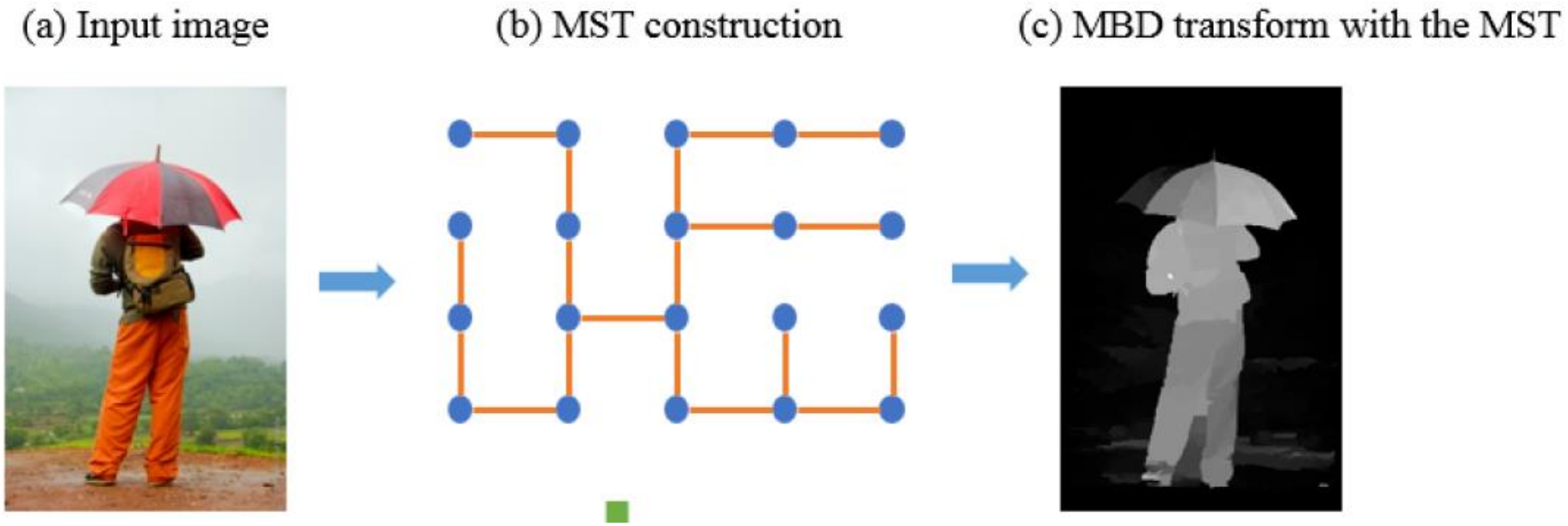
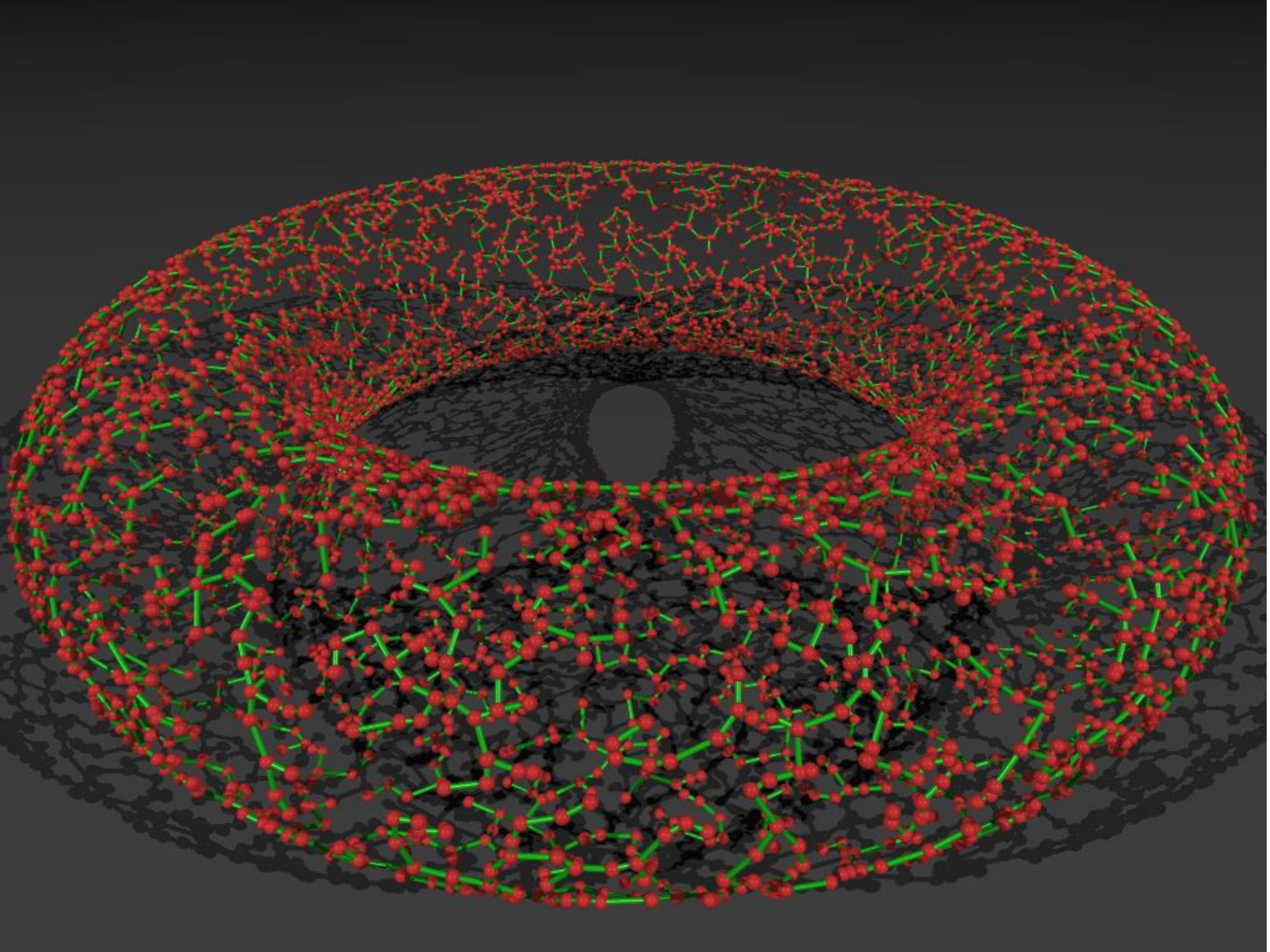
(a) (b) (c)

http://www.shengfenghe.com/uploads/1/5/1/3/15132160/cvpr16_mstsaliency.pdf

- „Naiv algoritmus”
  - Meghatározzuk minden pont távolságát a képszéli pontoktól (legrövidebb *súlyú* út a képszélhez)
  - A háttérpontok közelinek, az objektum határán belüliek pedig távolinak látszanak
    - Ha egyszínű hátterű kép közepében van egy másszínű kör, akkor minden háttér pont 0, minden körbeli pedig x távolságra lesz a képszéltől



- Javított algoritmus
  - A távolságot úgy értelmezzük, mint MST-beli távolság (*él-számban*)



Példa MST
a) Sötét kék: „seed-nodes” (képszéli pontok)
b) bottom-up átjárás
c) top-down átjárás
A körök a legközelebbi seed-node-tól mért távolságot tárolják