OI-OPPA. European Social Fund

Prague & EU: We invest in your future.

Kirkpatrick's Planar Point Location

Lukáš Beran
FEL CTU PRAGUE
beranlu6@fel.cvut.cz

Presentation plan

- Motivation
- Algorithm steps
- Complexity
- Literature

Motivation

- Slab method
  - *$O(n^2)$ memory, $O(\log n)$ time*
- Monotone chain tree in planar subdivision
  - *$O(n^2)$ memory, $O(\log^2 n)$ time*
- Trapezoidal map
  - *$O(n)$ expected memory, $O(\log n)$ expected time*
  - *$O(n \log n)$ expected preprocessing time*
- Kirkpatrick's Planar point location
  - *$O(n)$ memory, $O(\log n)$ time*

Algorithm steps

1. Data preprocessing
2. Building structure
3. Point query

Data preprocessing 1

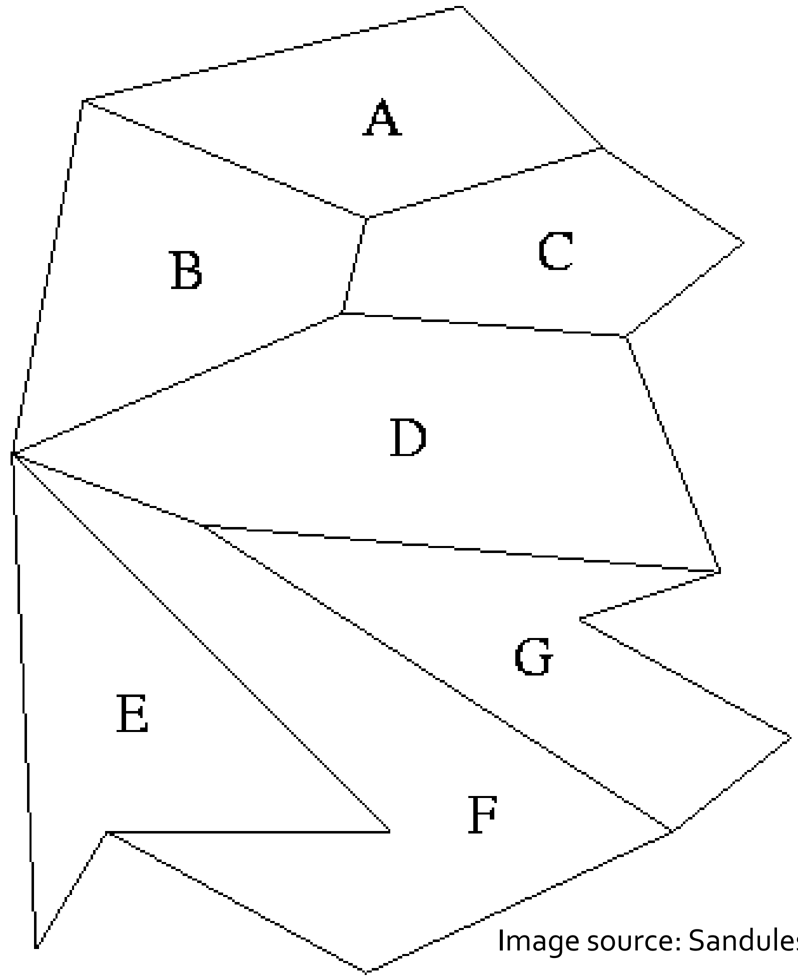


Image source: Sandulescu

Data preprocessing 2

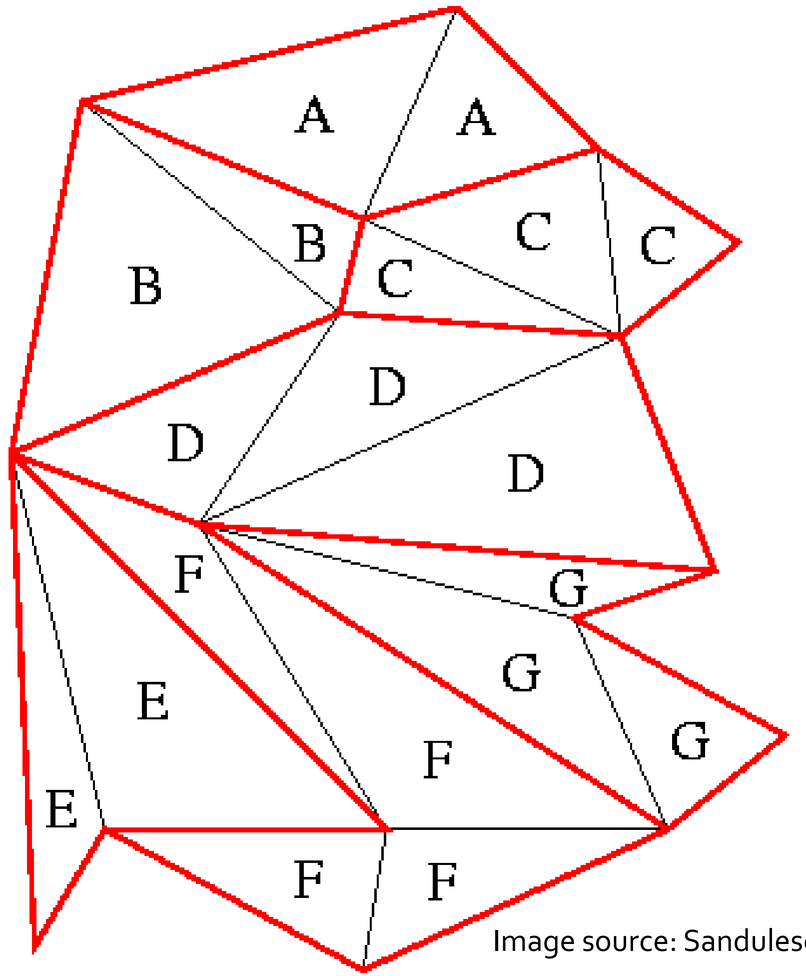


Image source: Sandulescu

Data preprocessing 3

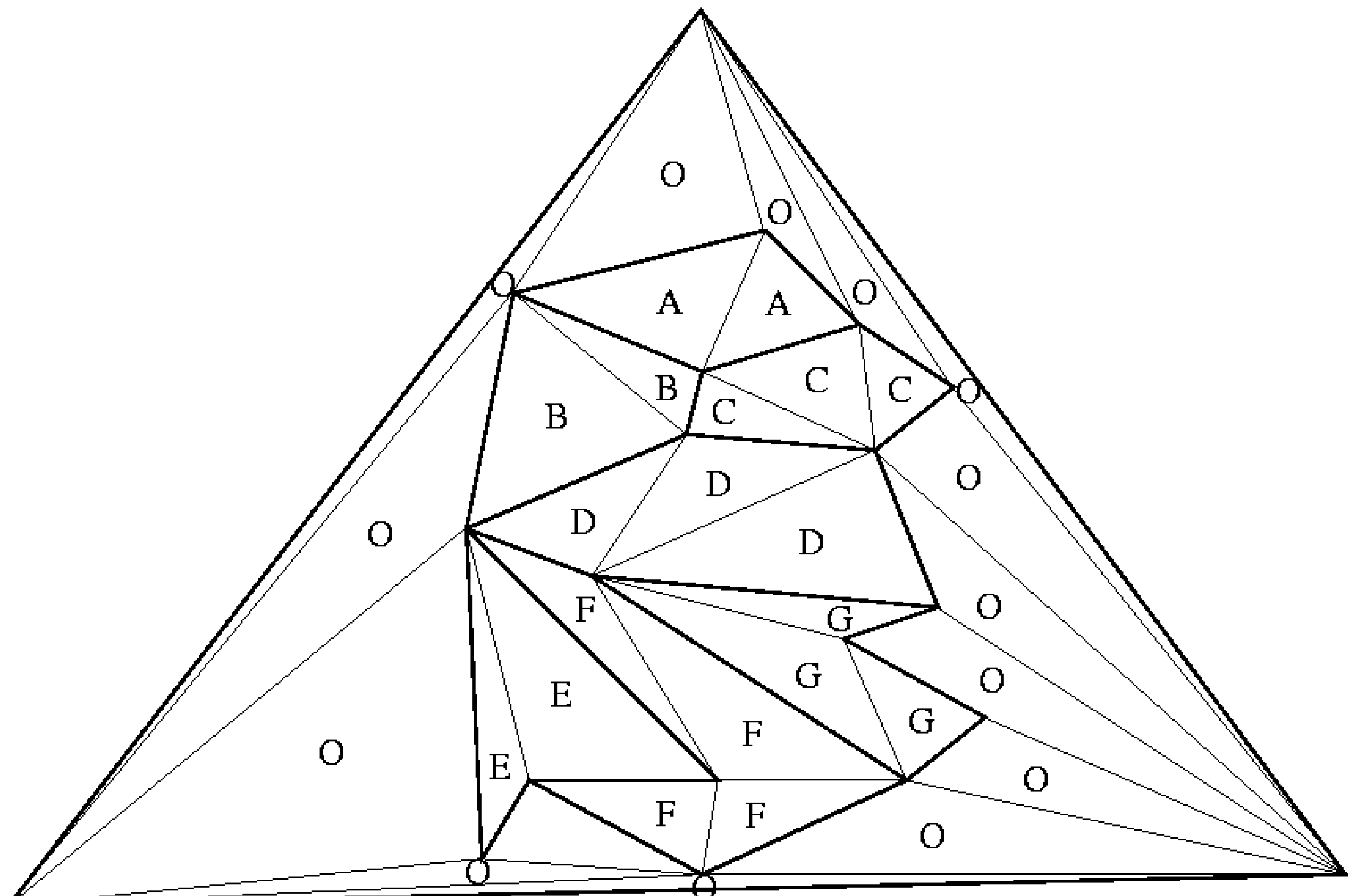


Image source: Sandulescu

Data preprocessing 4

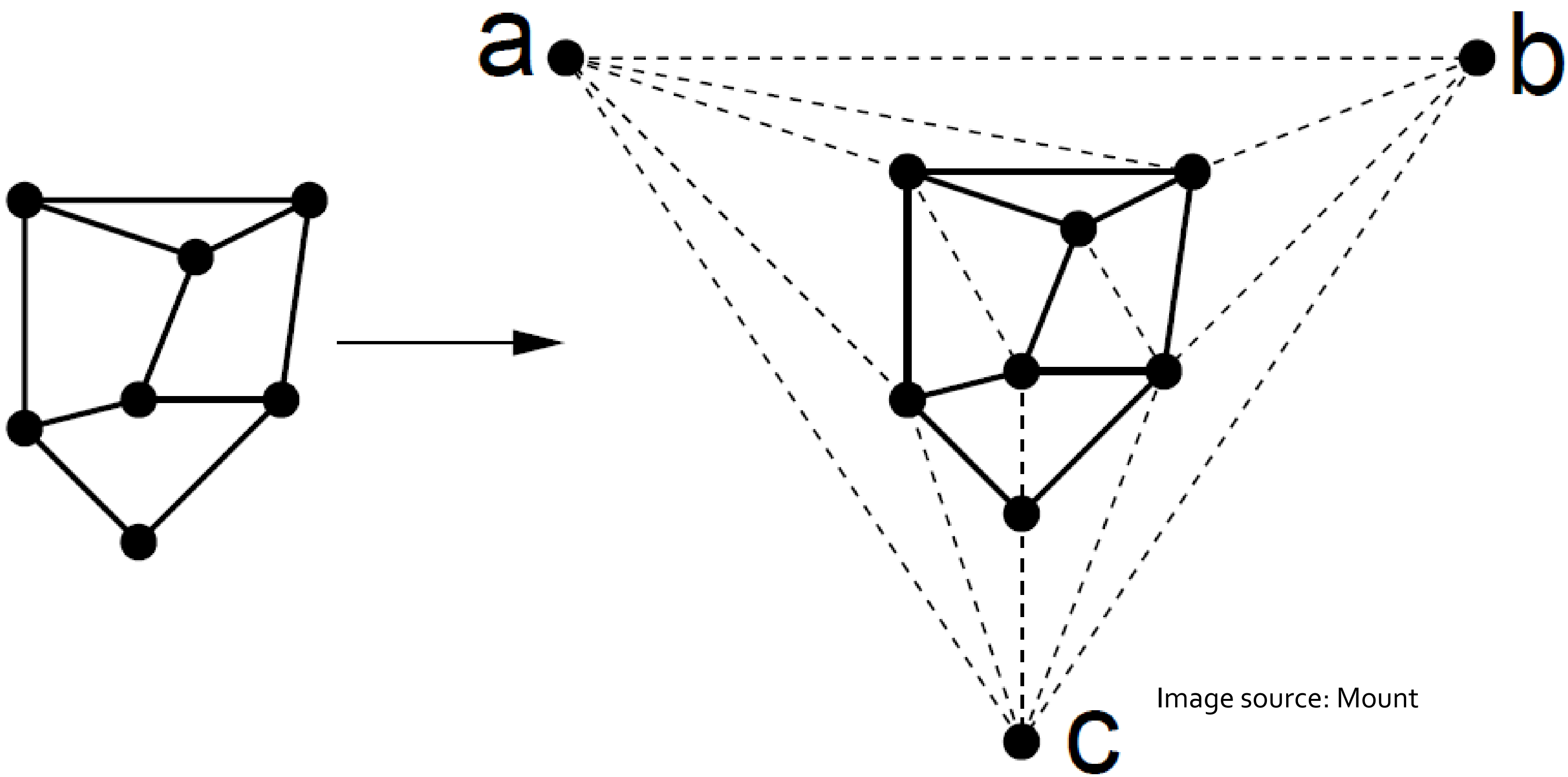


Image source: Mount

Data preprocessing 5

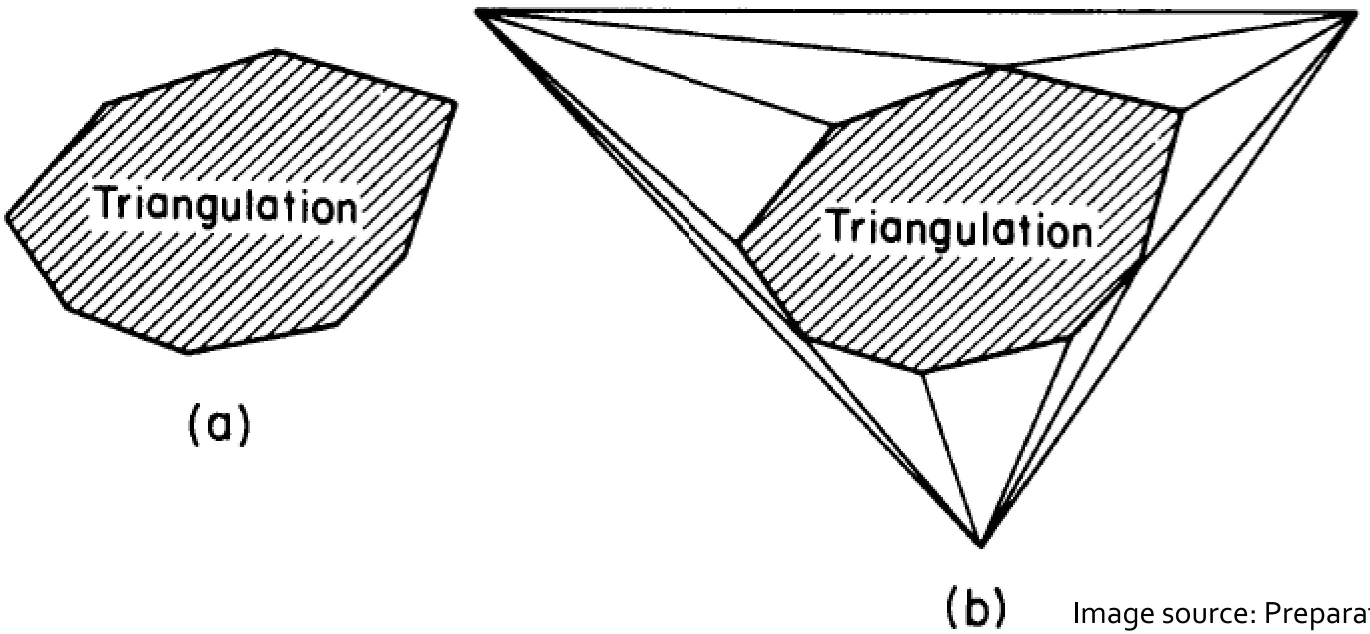


Image source: Preparata

Data preprocessing - summary

- Assumption that planar subdivision is a triangulation.
- If not, triangulate each face and label each triangular face with the same label as the original face.
- Compute the convex hull and triangulate the holes between the subdivision and CH.
- Put a large triangle around the subdivision and connect its vertices with CH.

Building structure 1

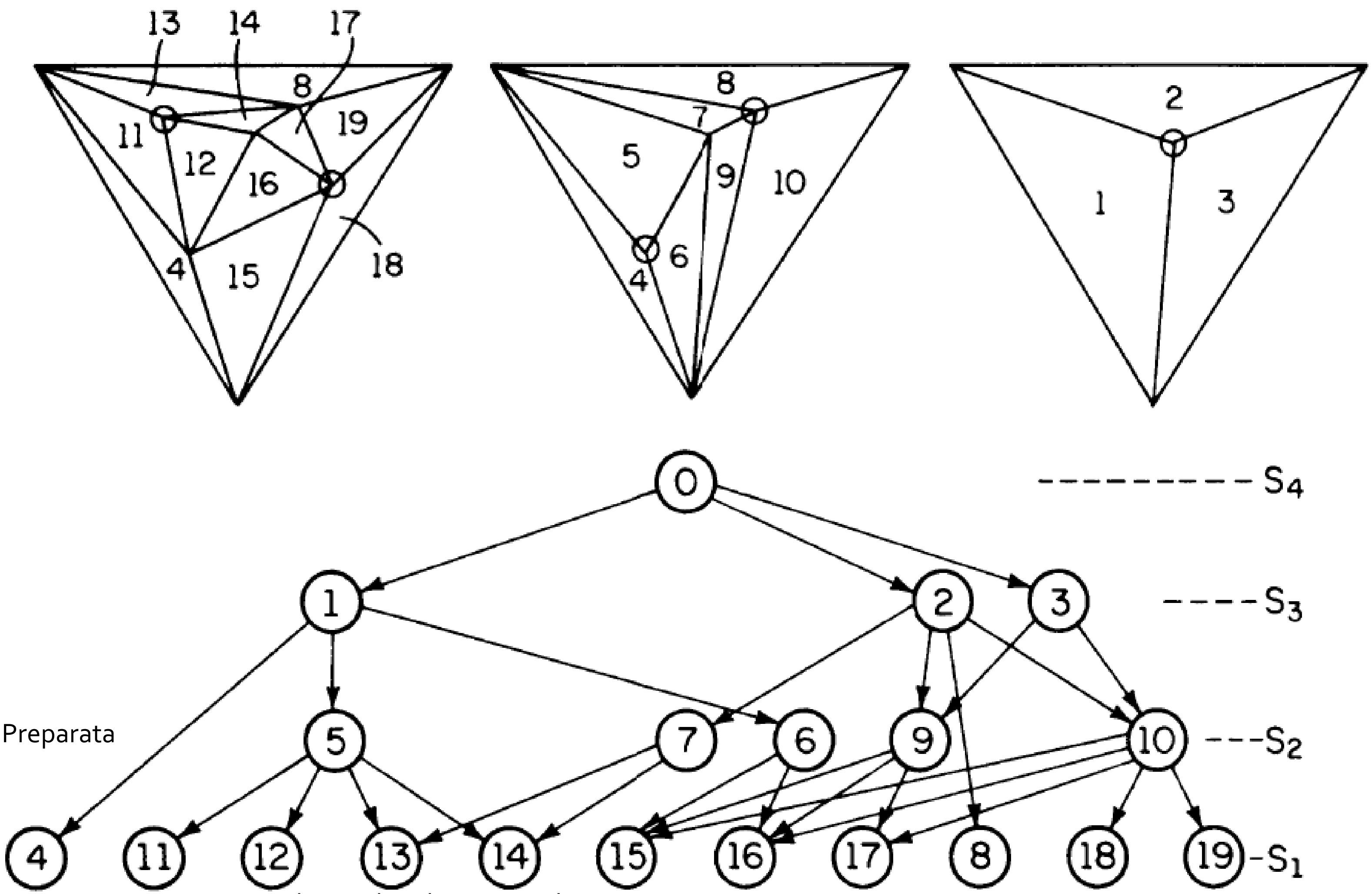


Image source: Preparata

Building structure2

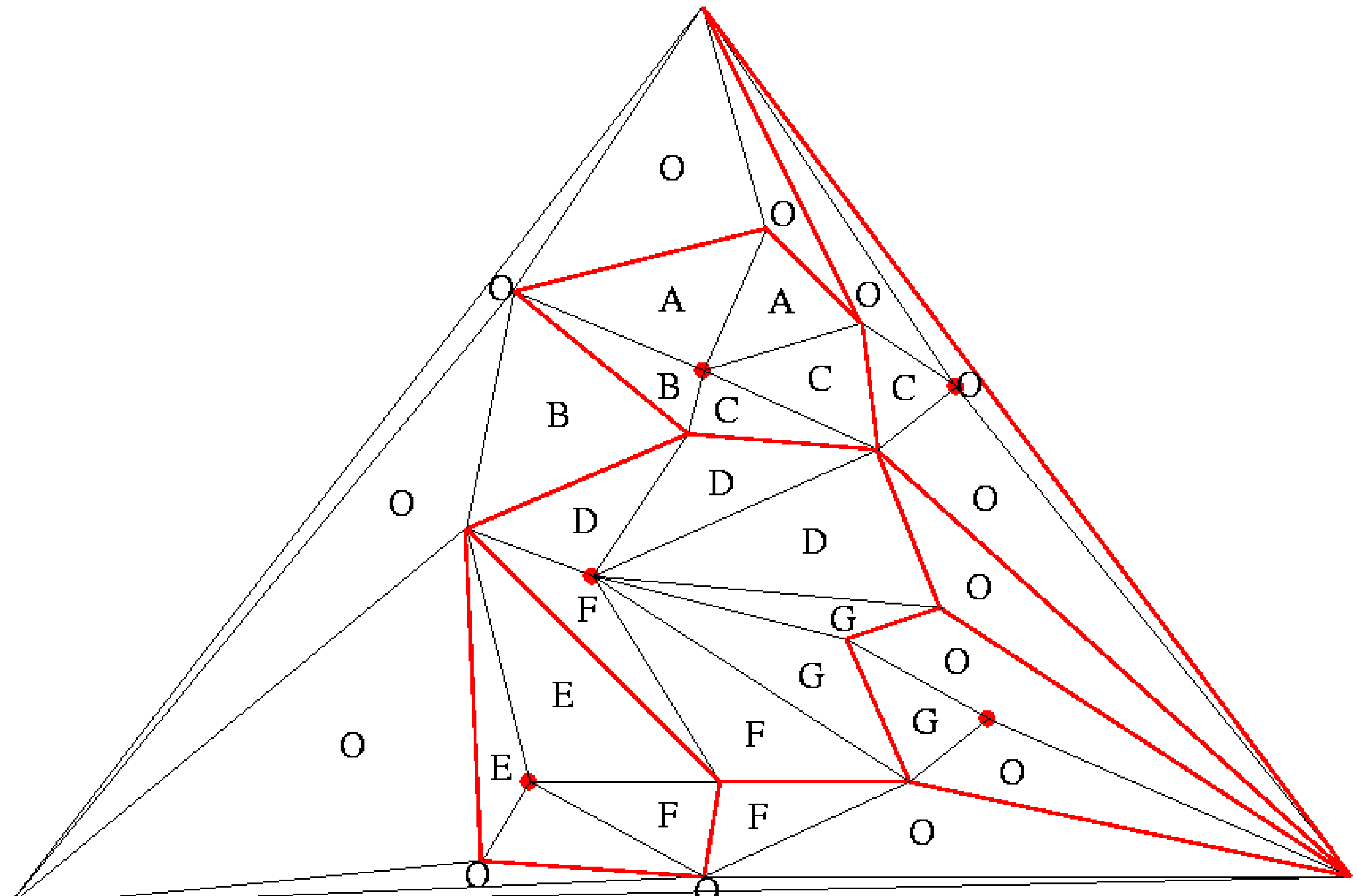


Image source: Sandulescu

Building structure 3

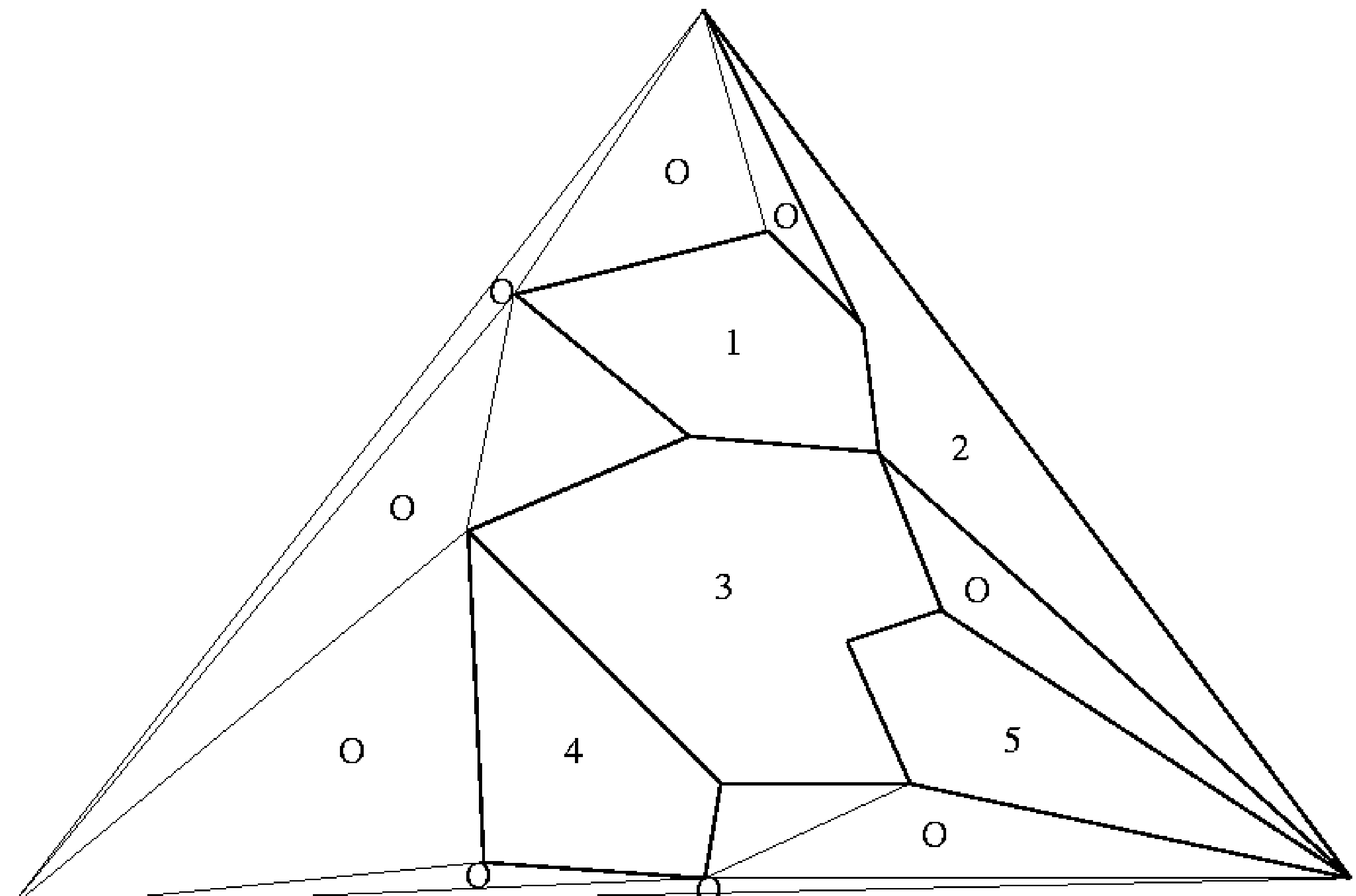


Image source: Sandulescu

Building structure 4

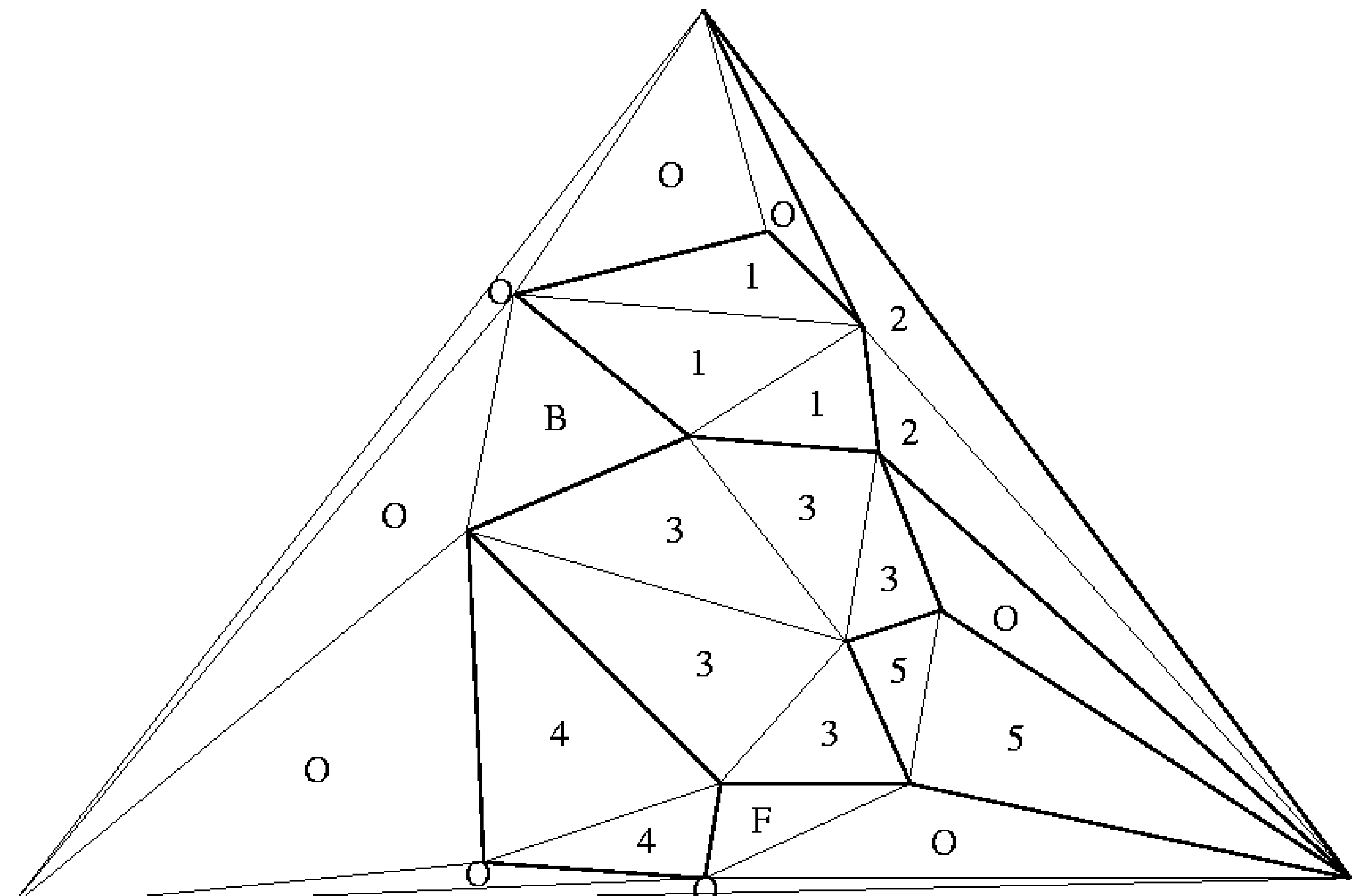


Image source: Sandulescu

Building structure 5

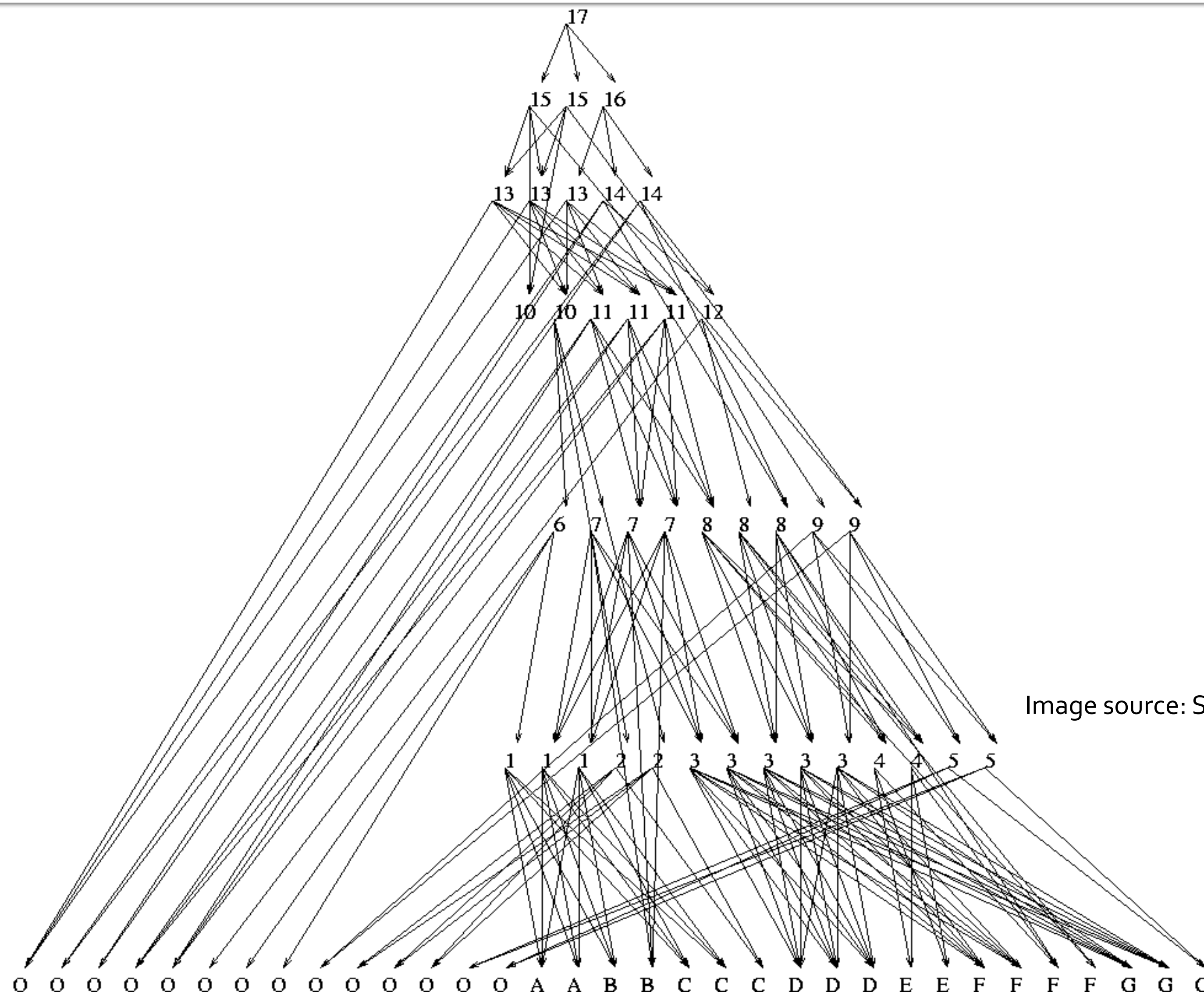


Image source: Sandulescu

Building structure 6

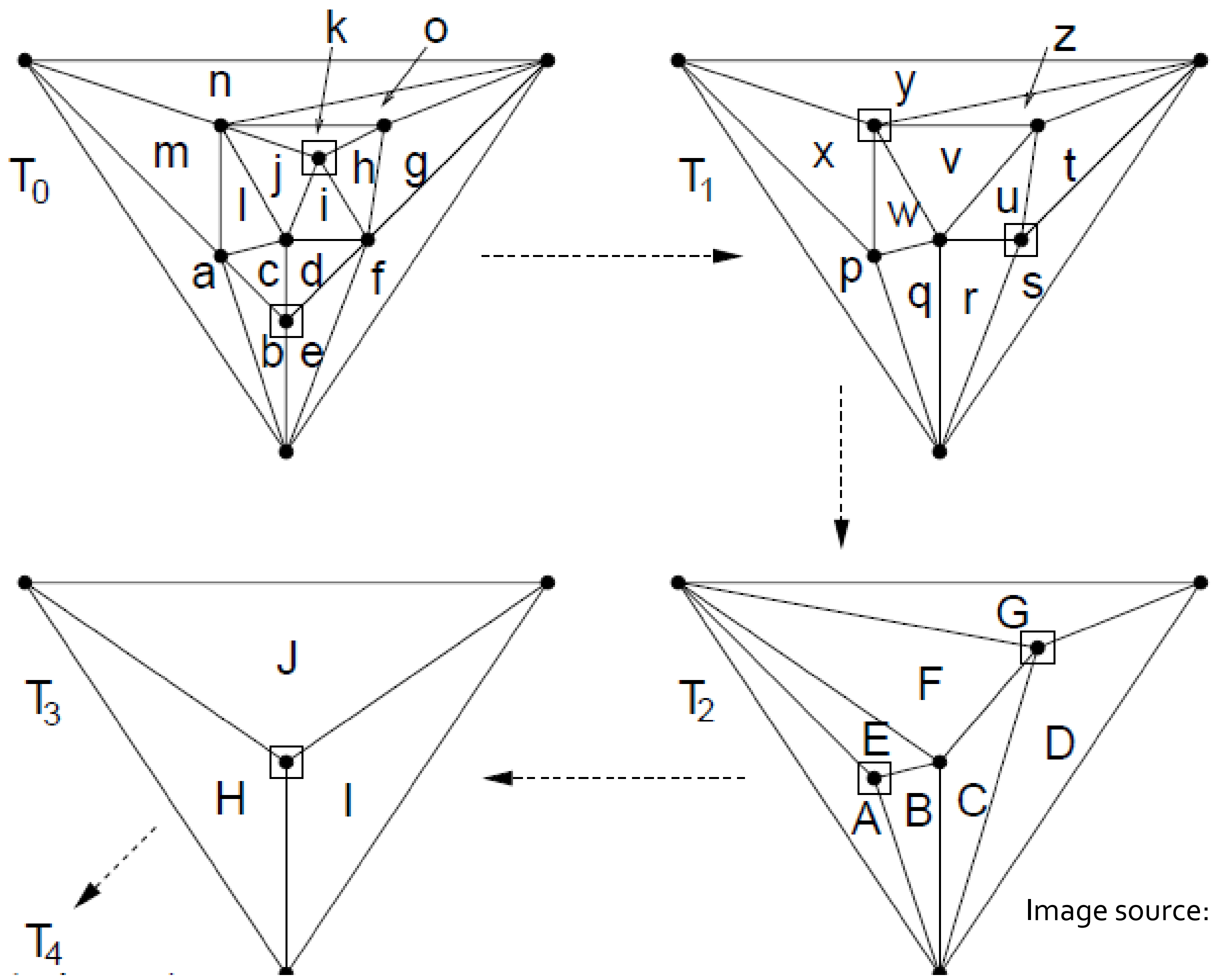


Image source: Mount

Building structure 7

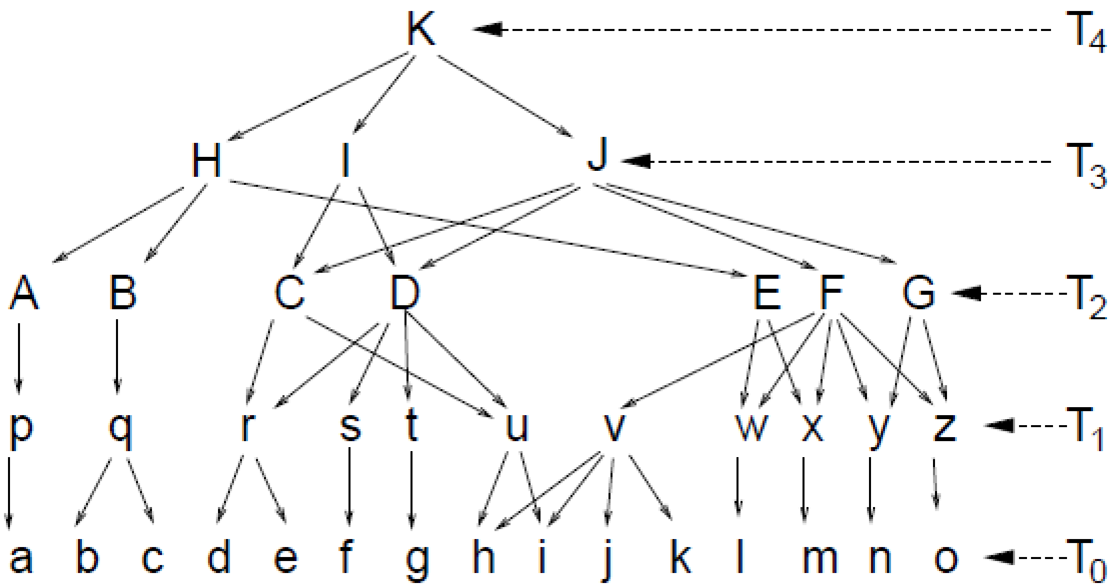


Image source: Mount

Building structure - summary

- Find an independent set of vertices with degree less than or equal to 8.
- Remove them from the graph, obtaining independent holes.
- Retriangulate the holes.
- Repeat the above steps until you are left with 3 vertices (the large triangle).

Point query 1

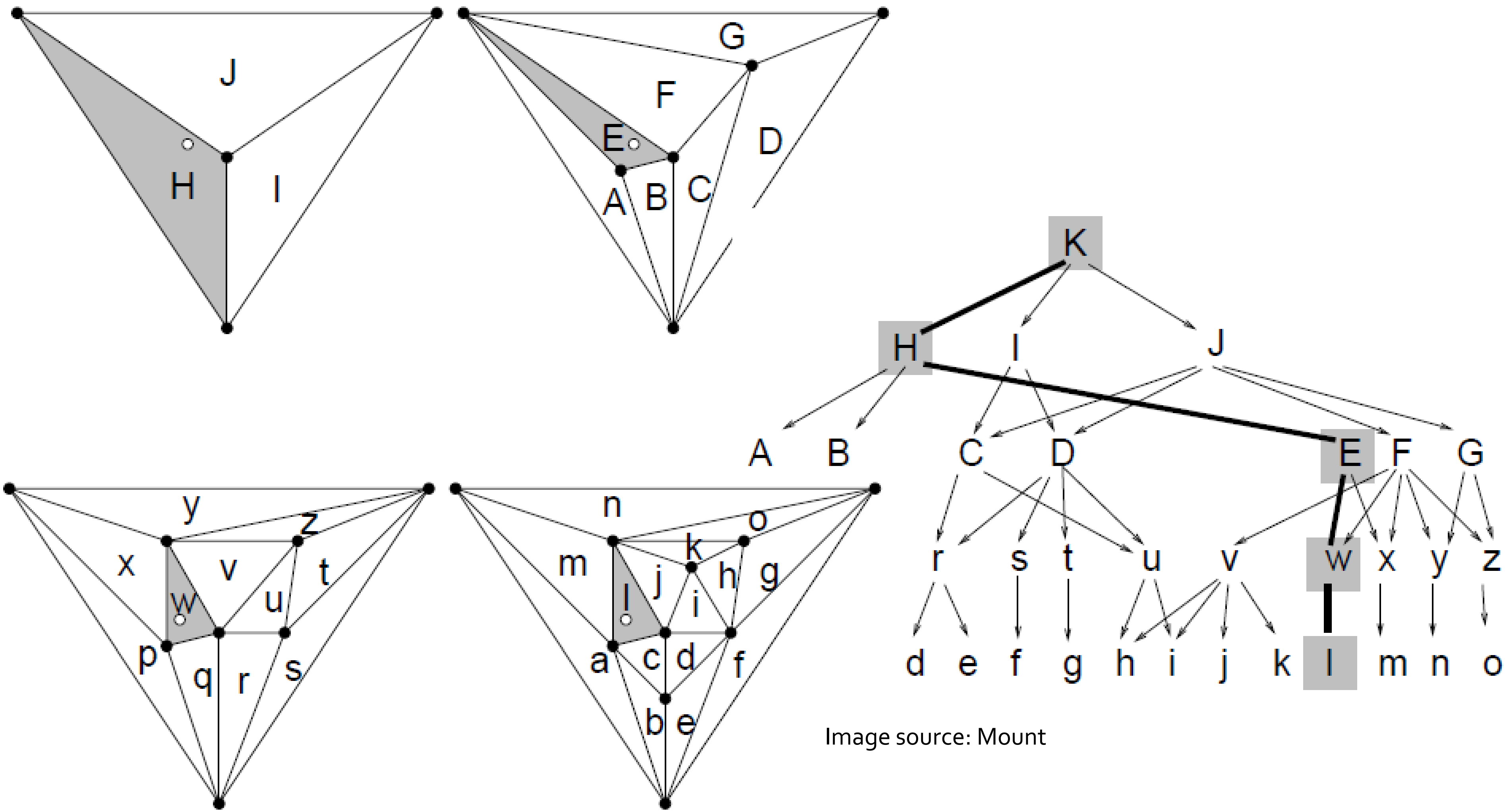


Image source: Mount

Point query 2

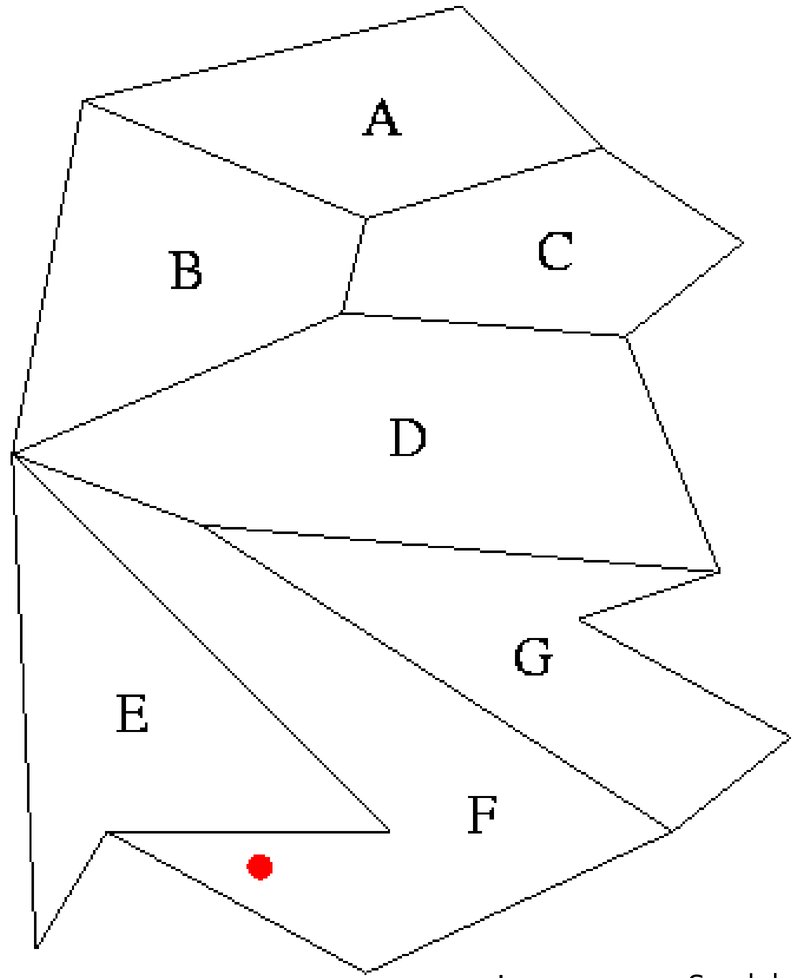


Image source: Sandulescu

Point query 3

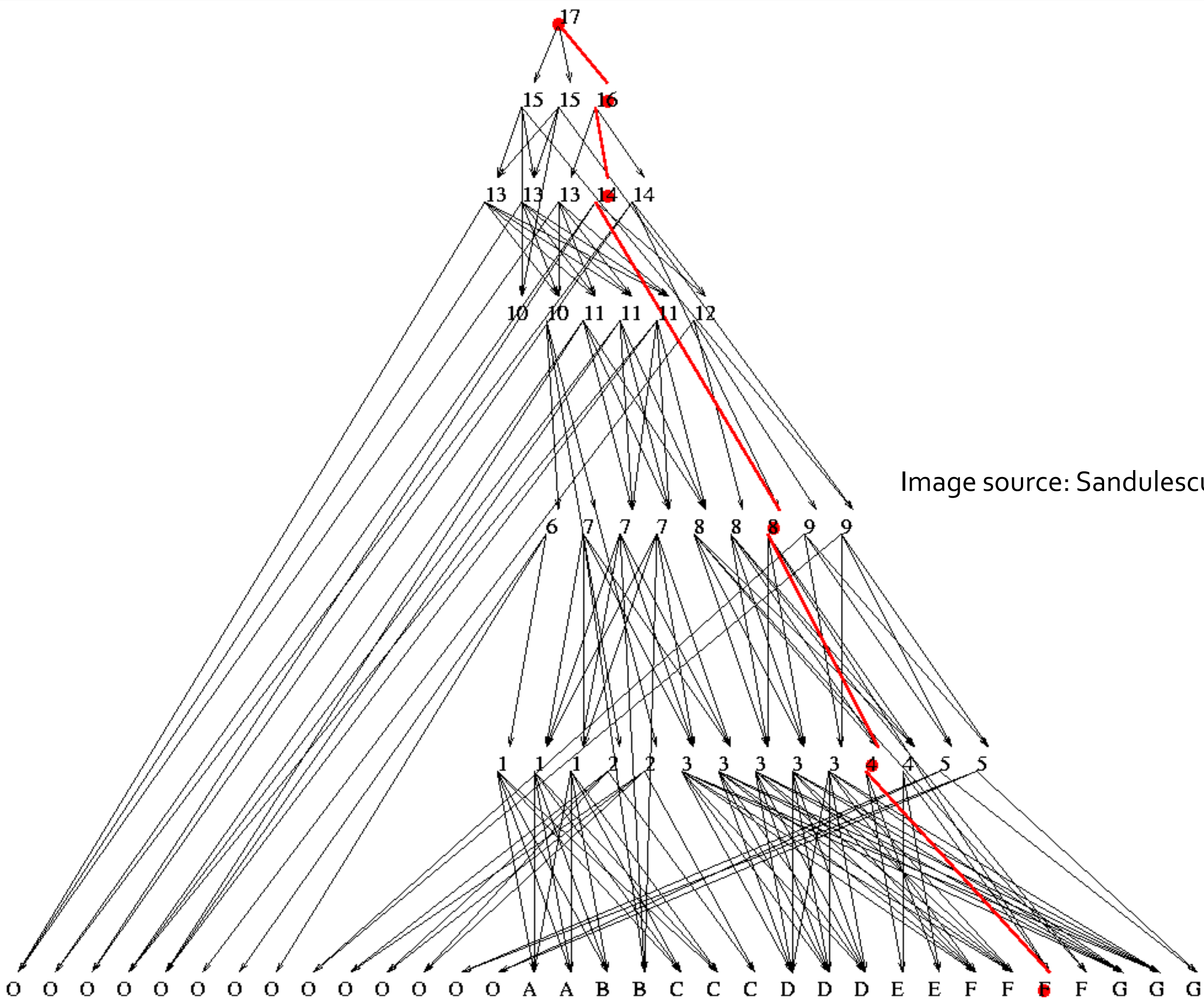


Image source: Sandulescu

Point query - summary

- Start in the root
- Find children node containing the point
- Continue from that node to leaf
- Point location algorithm in pseudo code :

procedure POINT-LOCATION
begin if ($z \notin$ TRIANGLE(root)) **then** print "z belongs to unbounded region"
else begin v := root;
while ($\Gamma(v) \neq \emptyset$) **do**
for each $u \in \Gamma(v)$ **do if** ($z \in$ TRIANGLE(u)) **then** $v := u$;
print v
end
end.

Image source: Preparata

Complexity

- Lemma: Every planar graph on n vertices contains an independent vertex set of size 1/18n in which each vertex has degree at most 8. The set can be found in O(n) time.
- Layer T+1 has at most 17/18n vertices of layer T.
- depth = $\log_{18/17} n \approx 12 \log n$
- Time complexity is O(log n)

Complexity

- Space complexity = sum up the sizes of triangulations.
- $n(1+(17/18)+(17/18)^2+(17/18)^3+\ldots \leq 18n$
- (sum of geometric series : $S = a_1 / 1 - q$)
- Space complexity is $O(n)$

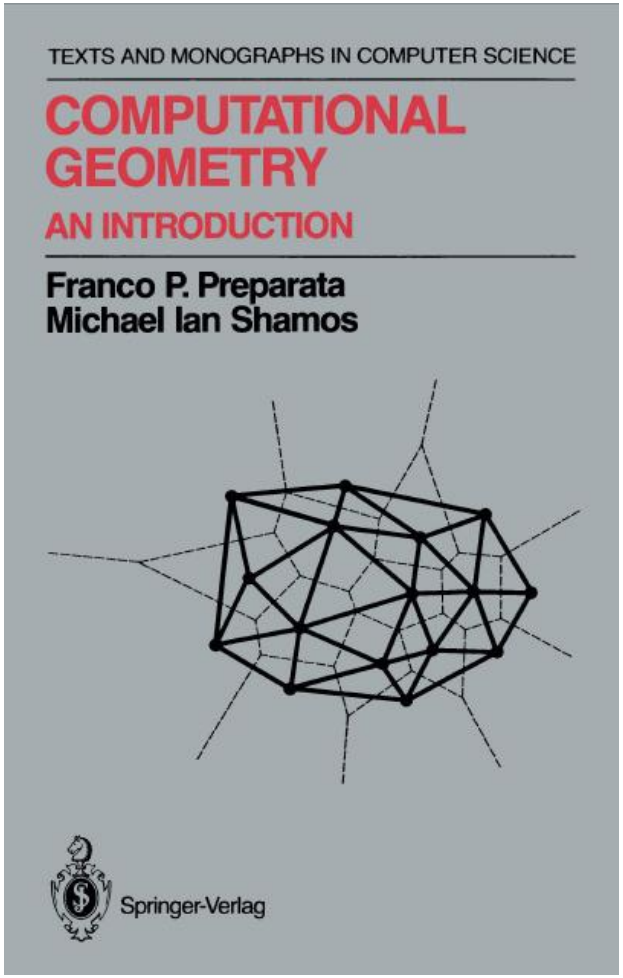
Summary

- Very good time and space O complexity
- Big multiplicative constants – time 12*log(n), space 18*n
- Trapezoidal map is more simple to implement and often is faster then Kirkpatrick planar location

Literature

- Mount, D.: Computational Geometry Lecture Notes for Spring 2007
- Franco P. Preparata, Michael I. Shamos: Computational Geometry: An Introduction, 1985
- *Subhash Suri:* Point Location, http://www.cs.ucsb.edu/~suri/cs235/Location.pdf
- Sandulescu, P.:Kirkpatrick's Point Location Data Structure, http://cgm.cs.mcgill.ca/~athens/cs507/Projects/2002/PaulSandulescu/index.html

Computational geometry



Thank you for your attention

Time for discussion

OI-OPPA. European Social Fund

Prague & EU: We invest in your future.