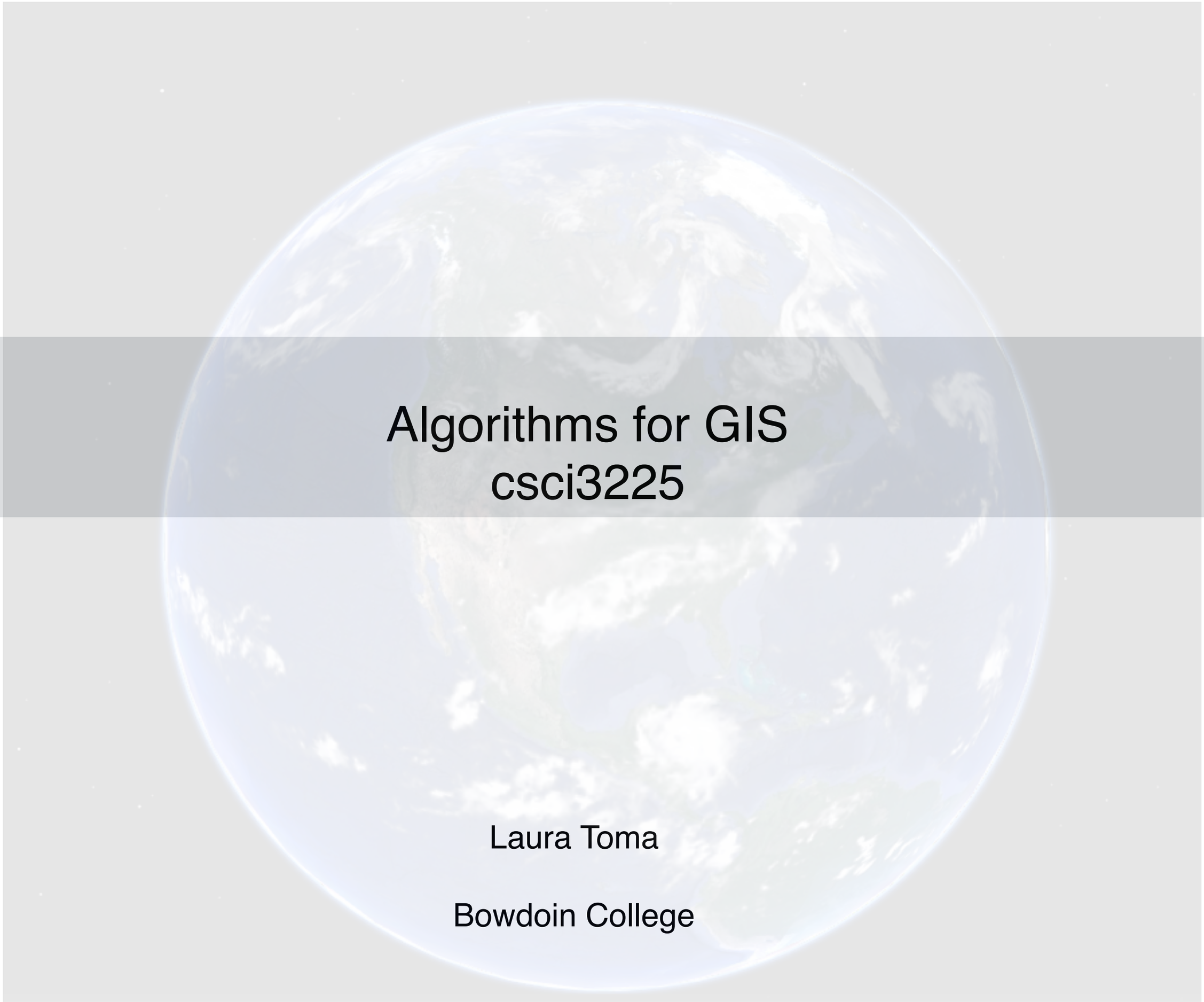


Algorithms for GIS
csci3225

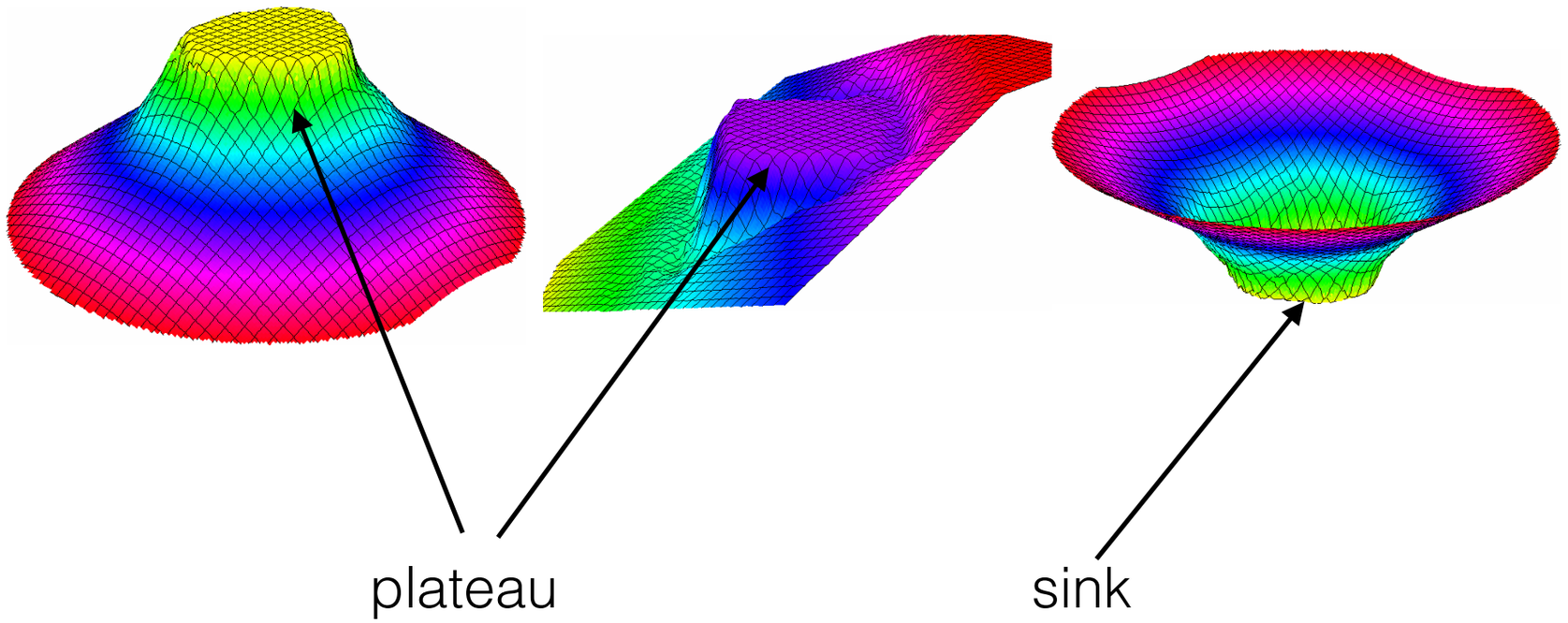
Laura Toma

Bowdoin College

Flow on digital terrain models (III)

- FD in sinks via sink filling

Computing FD on flat areas



plateau

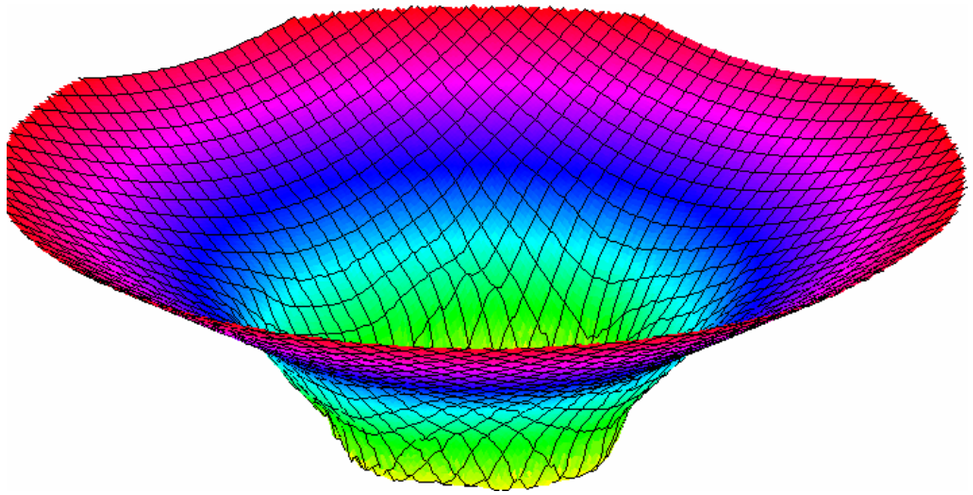
flat area has at least one spill point from which water flows downhill

sink

flat area surrounded by higher grounds

FD in sinks

- **Downslope FD not possible**



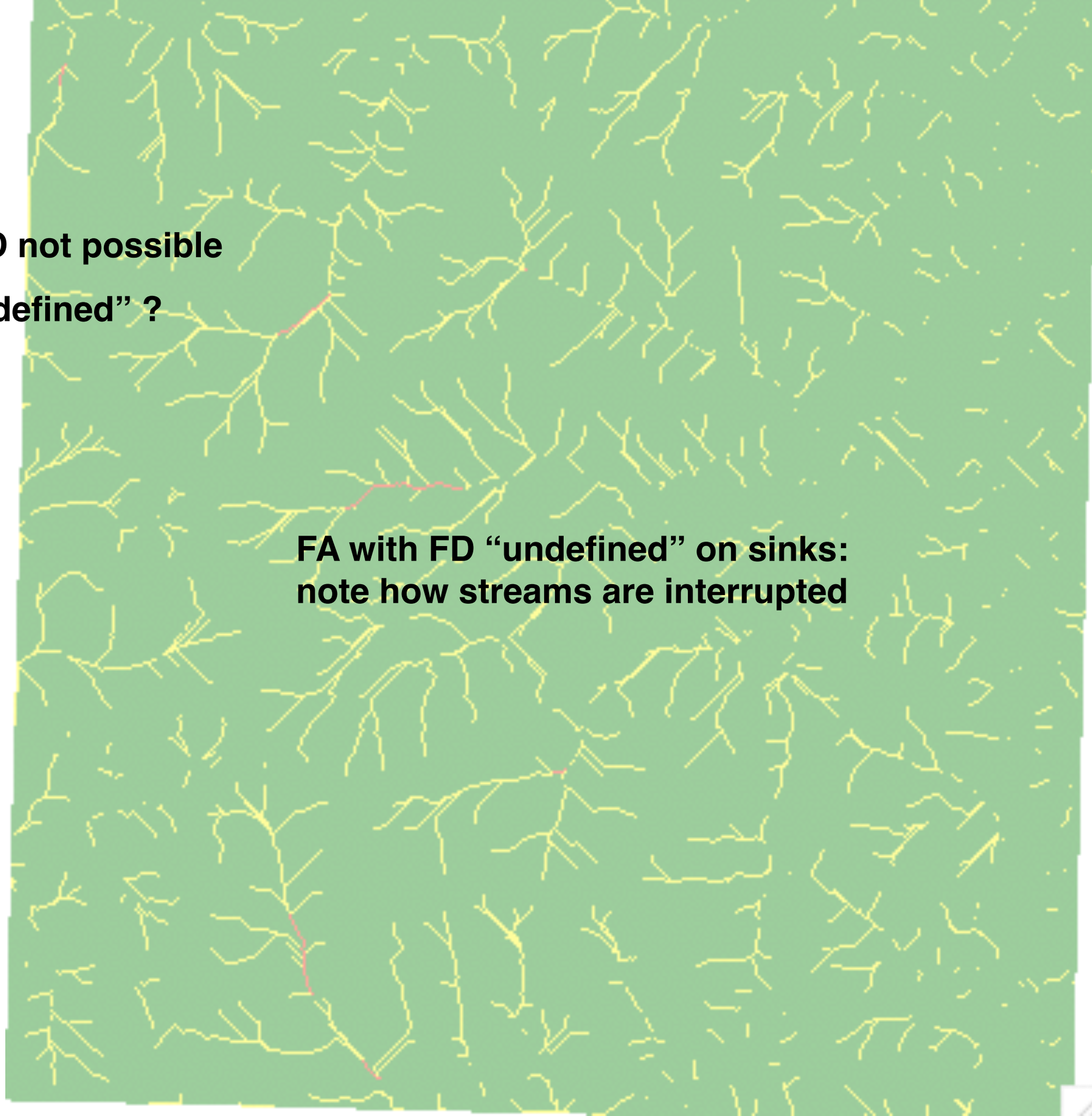
Water comes into the sink from the surrounding area and cannot go “out” following downslope paths

FD in sinks

- **Downslope FD not possible**
- **Leave FD “undefined” ?**

FD in sinks

- **Downslope FD not possible**
- **Leave FD “undefined” ?**



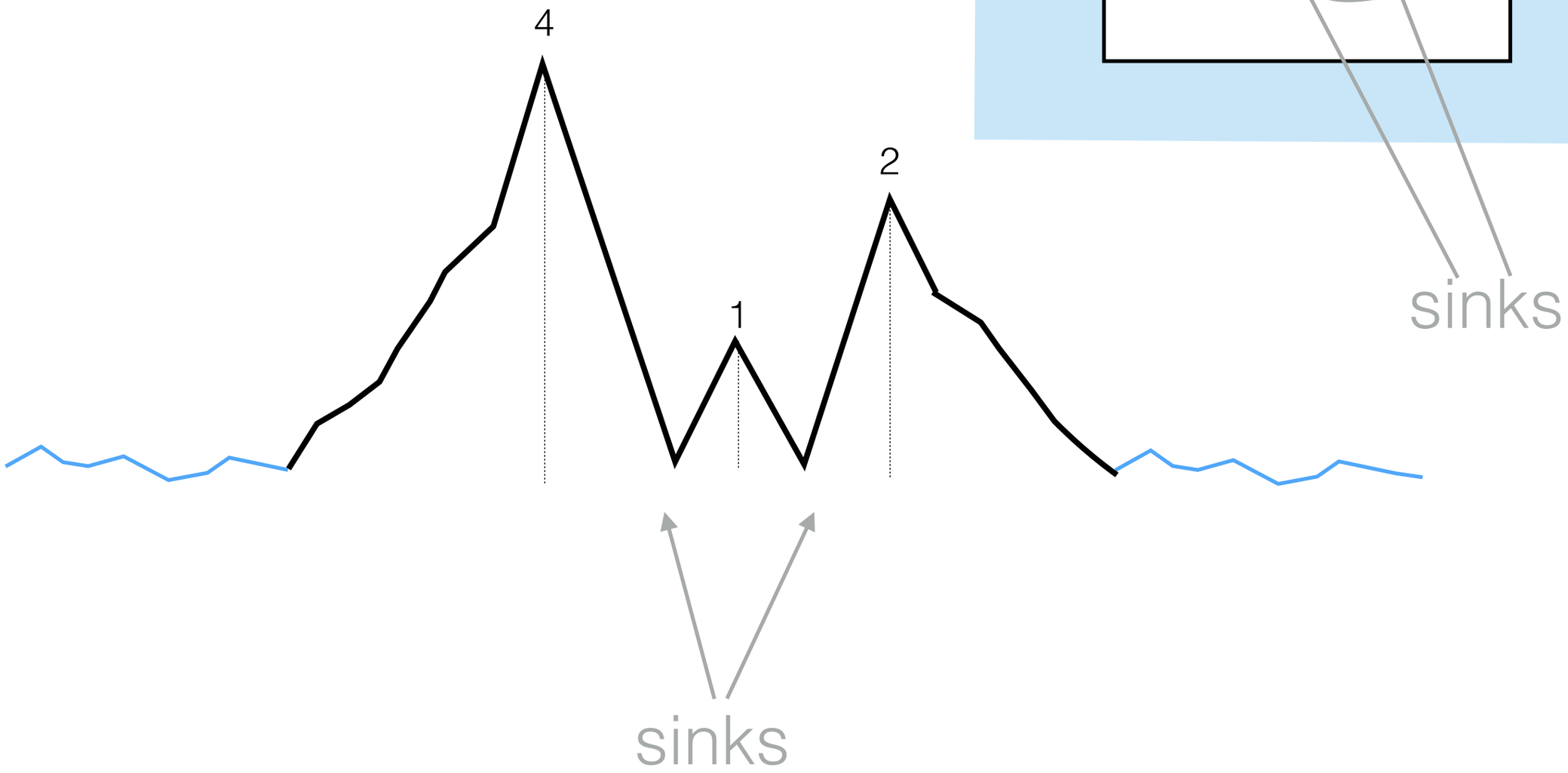
FA with FD “undefined” on sinks:
note how streams are interrupted

FD in sinks

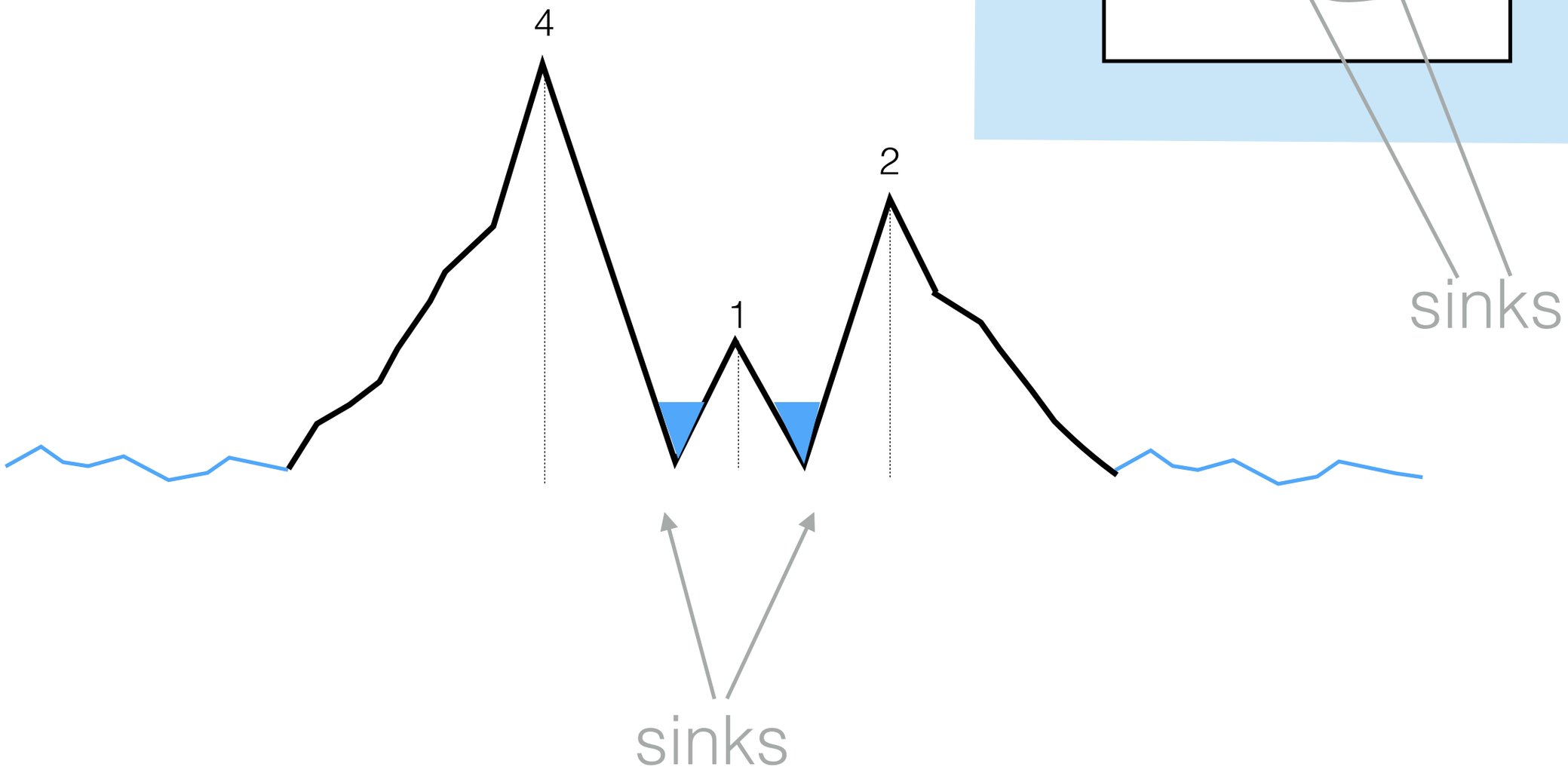
- **Downslope FD not possible**
- **Leave FD “undefined”** leads to unrealistic, interrupted rivers
- **Flooding the sinks**

    - Model flooding the terrain with a large enough amount of water (rain) that falls uniformly across the terrain. Assume the terrain borders the “ocean”.
    - At steady state: all points in the terrain have found a path towards the ocean, and more rain does not raise the water level.

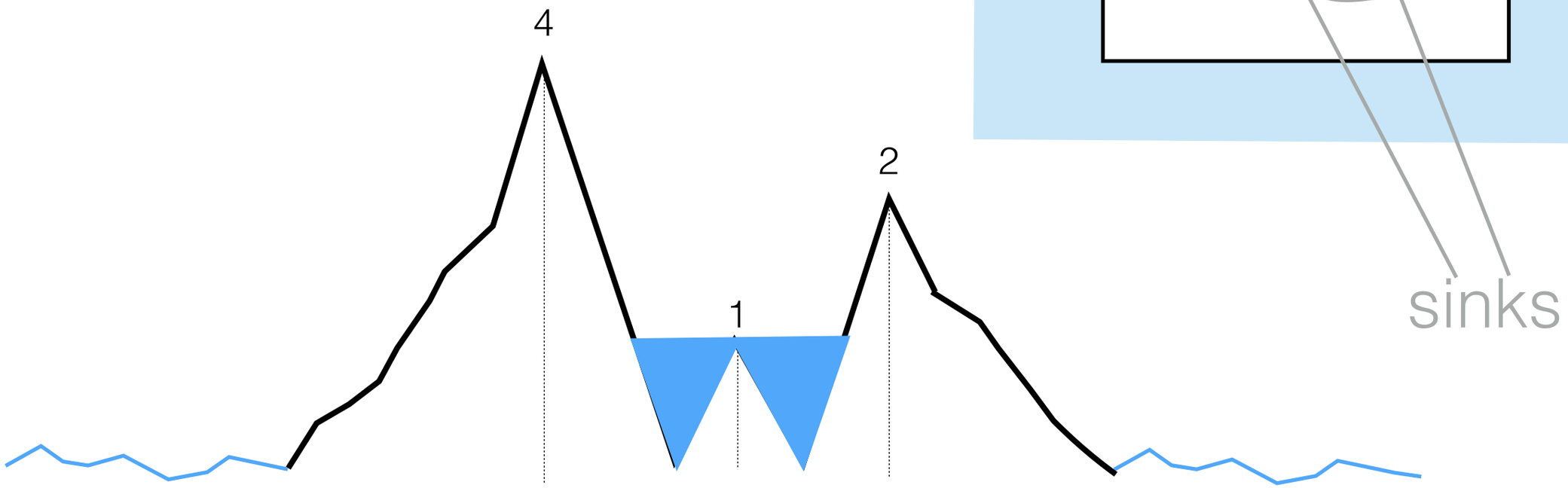
Flooding the sinks



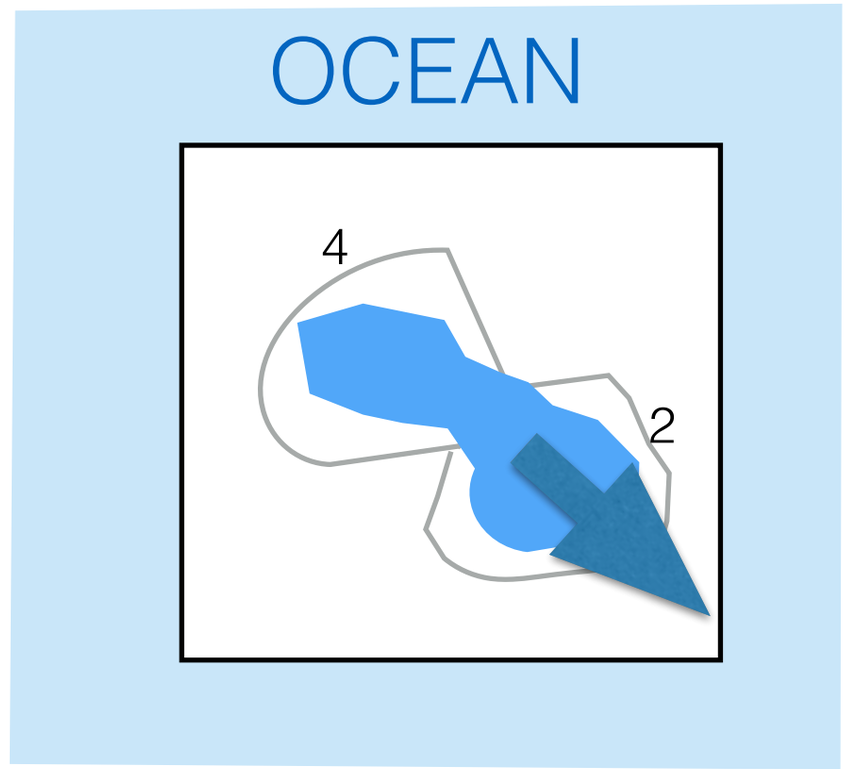
Flooding the sinks

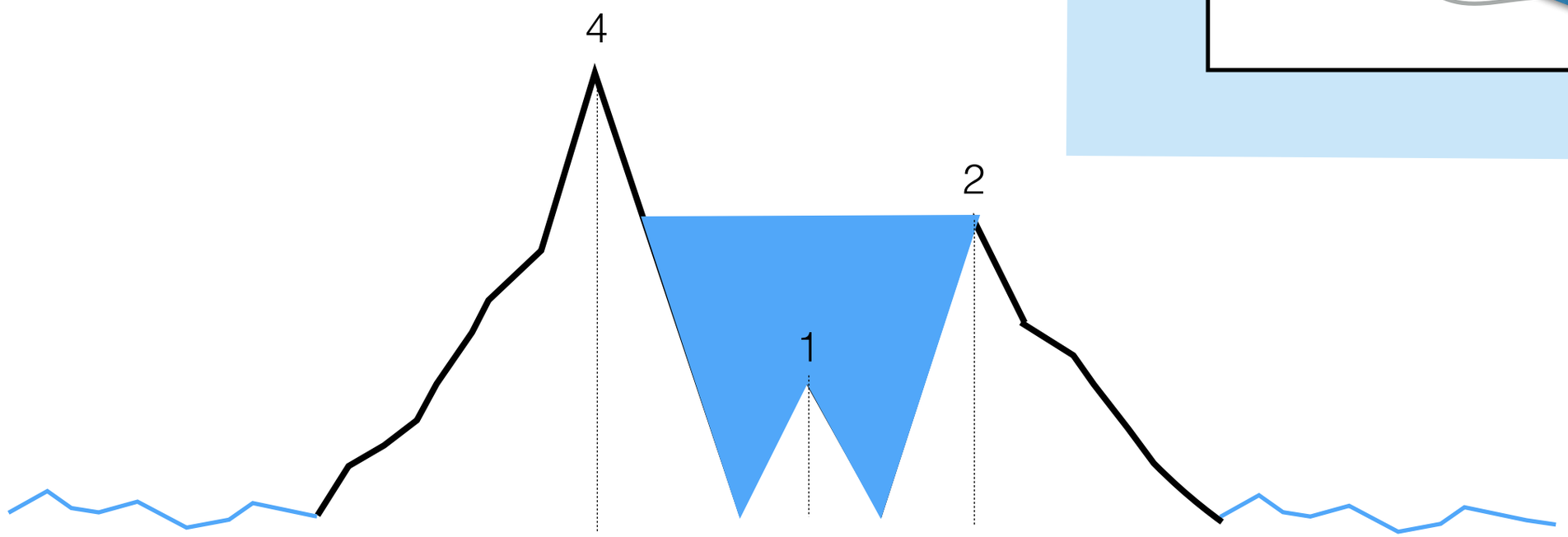


Flooding the sinks

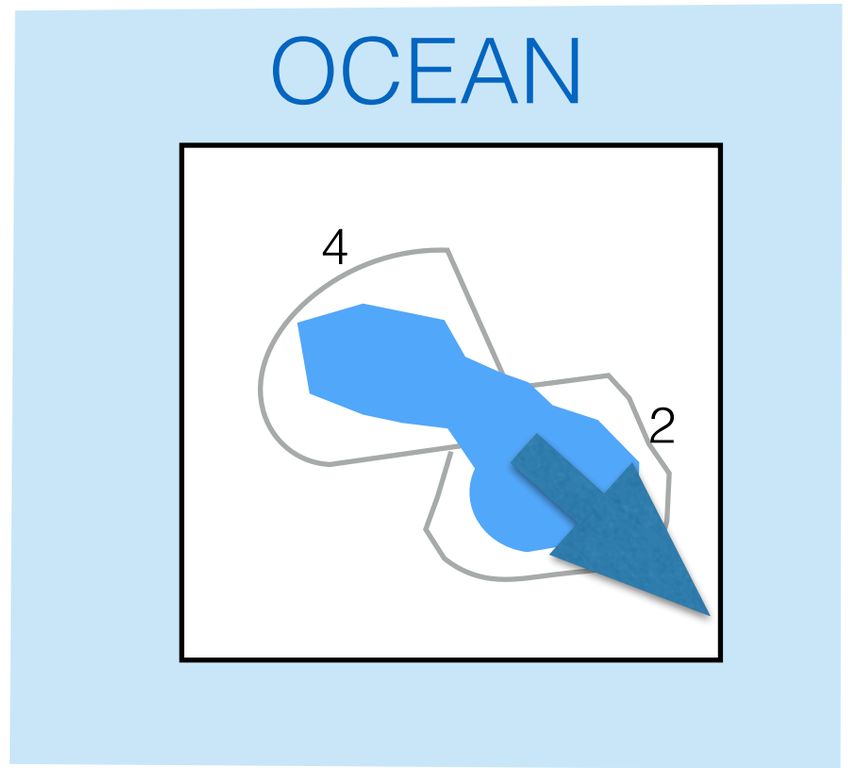


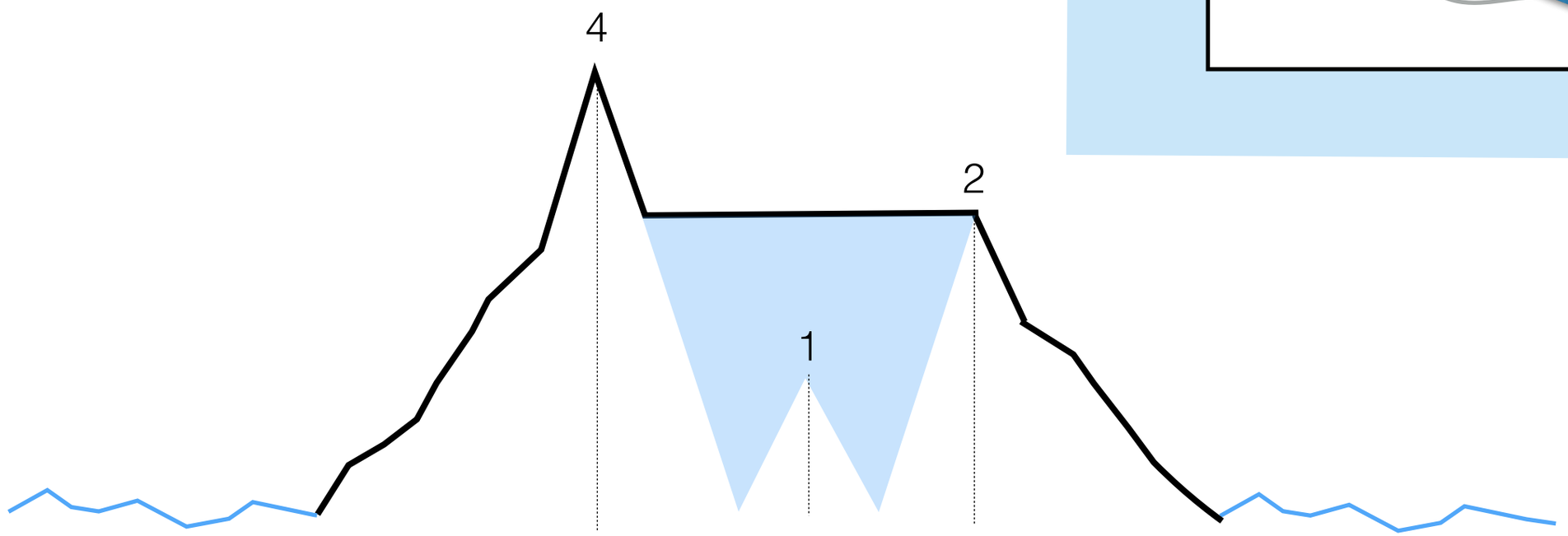
Flooding the sinks





Flooding the sinks





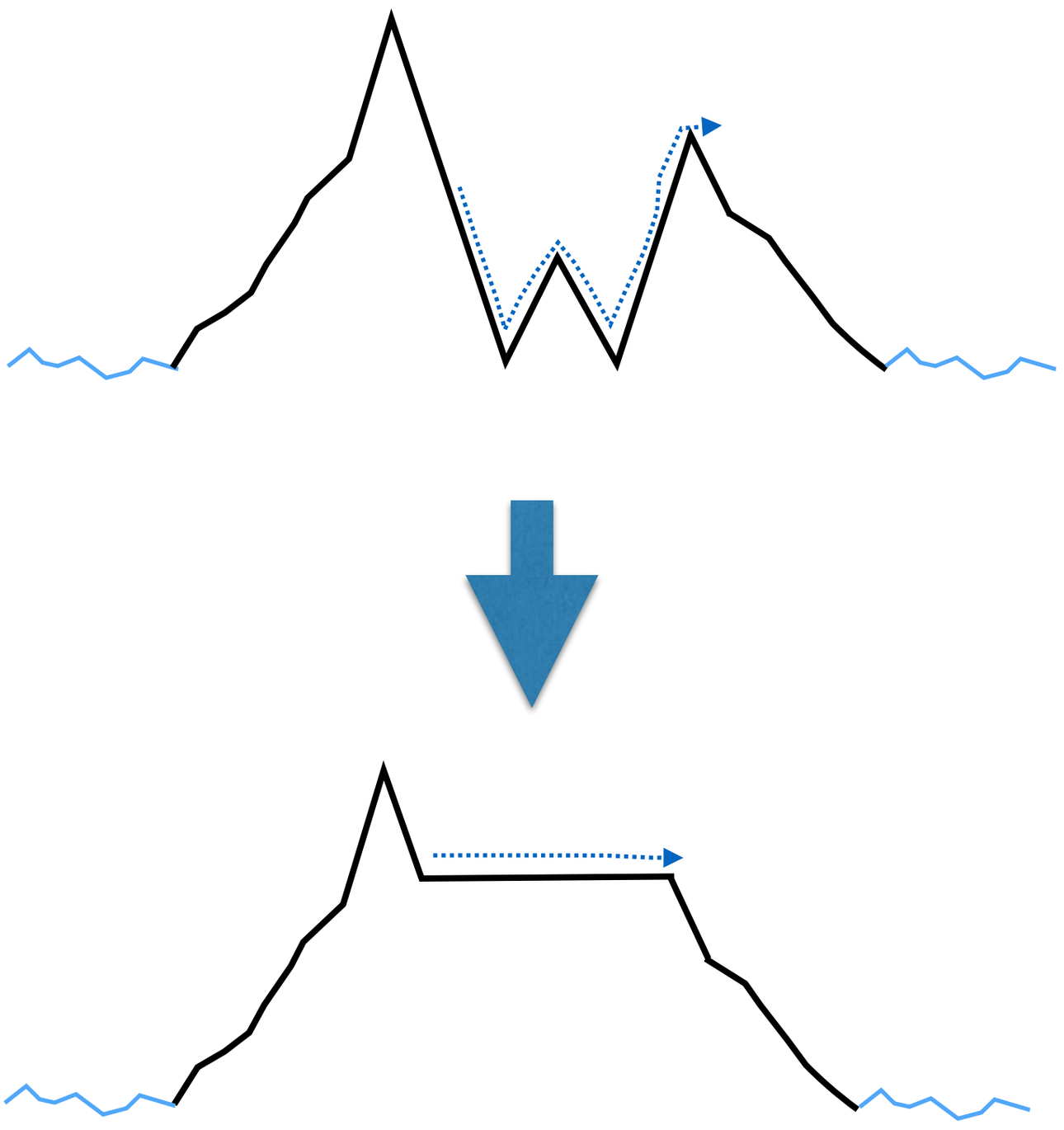
steady state

- **A flooded terrain has no sinks**: All its flat areas are plateaus.

=> We can compute FD at every point

Goal: Compute flooded terrain

original terrain

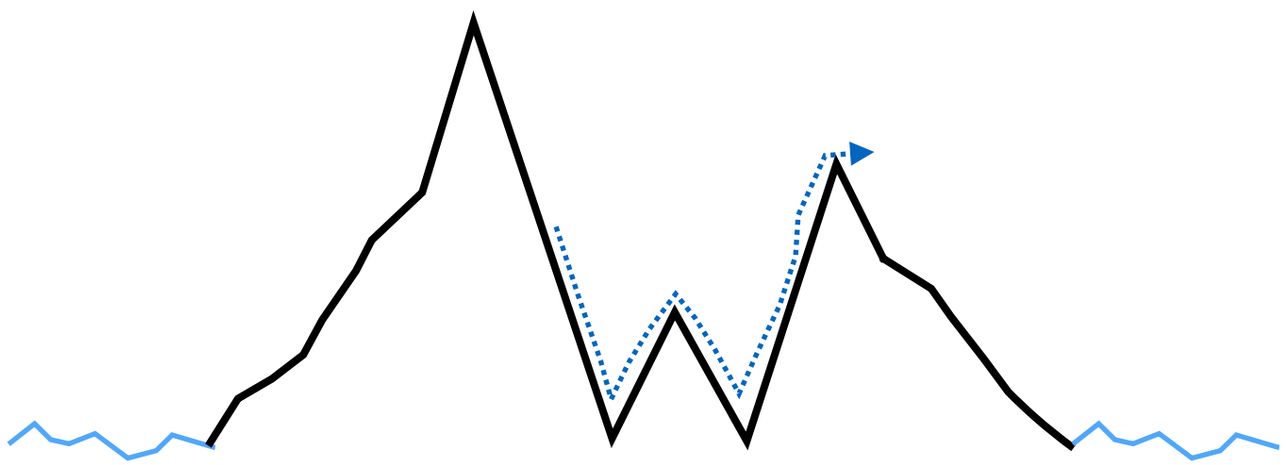


flooded terrain

Once we compute the flooded terrain, we can use it to compute FD

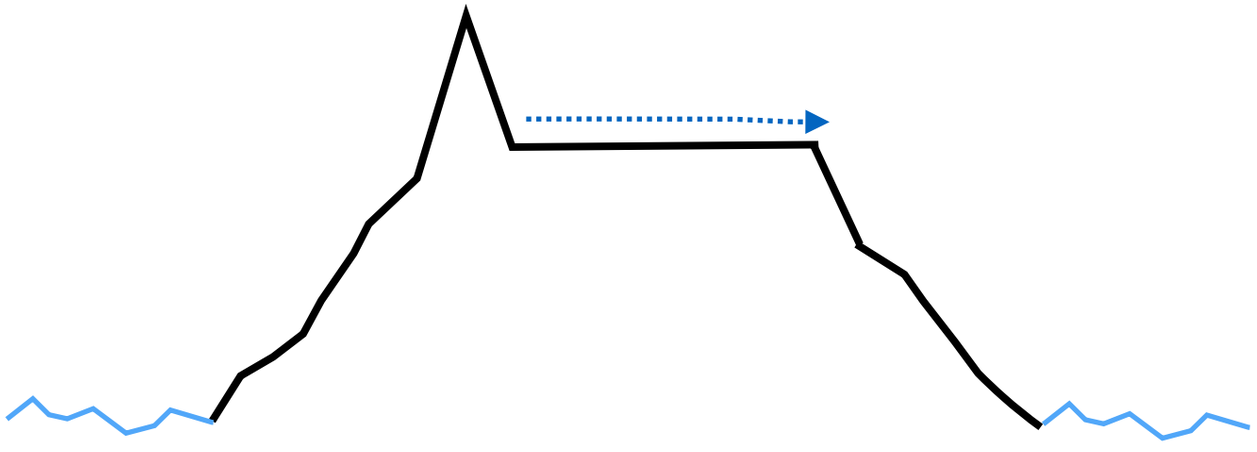
Goal: Compute flooded terrain

original terrain



HOW?

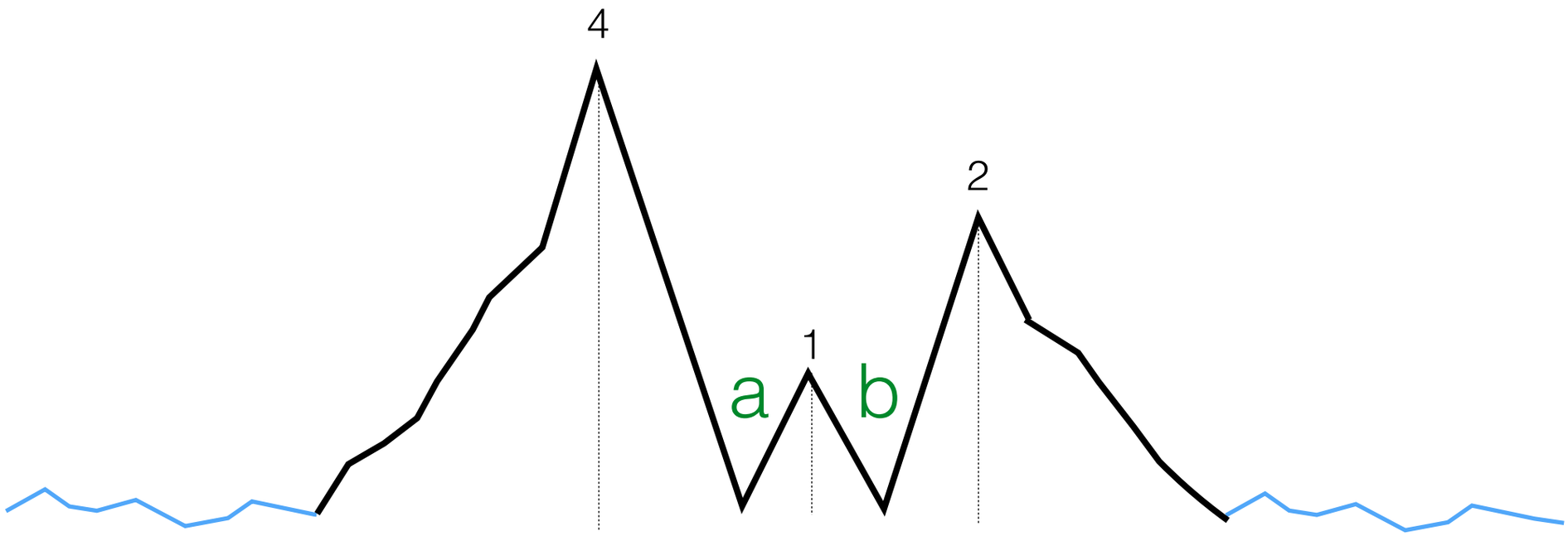
flooded terrain

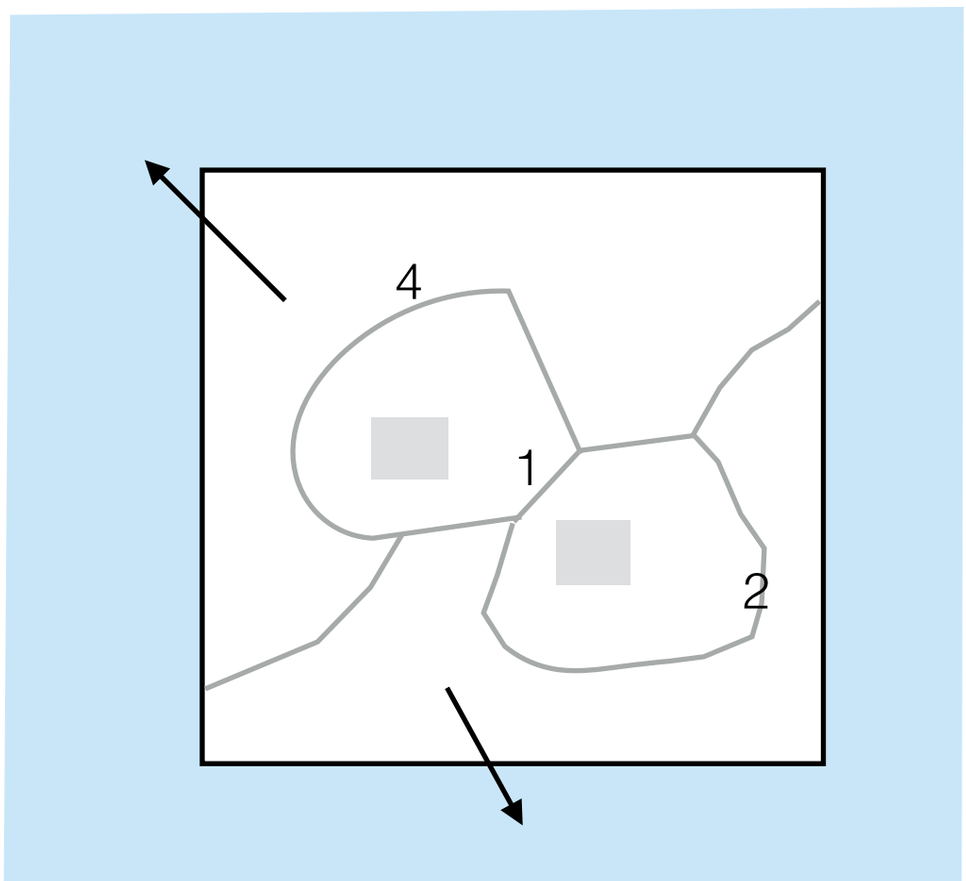


Once we compute the flooded terrain, we can use it to compute FD

Computing FD: The overall process

1. Compute FD
2. **Plateaus**: Identify plateaus and compute FD on plateaus
3. **Sinks**: Identify the sinks and their basins
4. **Flood**: simulate flooding the sinks, and, for each sink, record the lowest height at which it merges with the ocean.
5. **Raise**: For each sink, traverse its basin and raise each point that's below the flooded level to the flooded level
6. Compute FD on flooded terrain



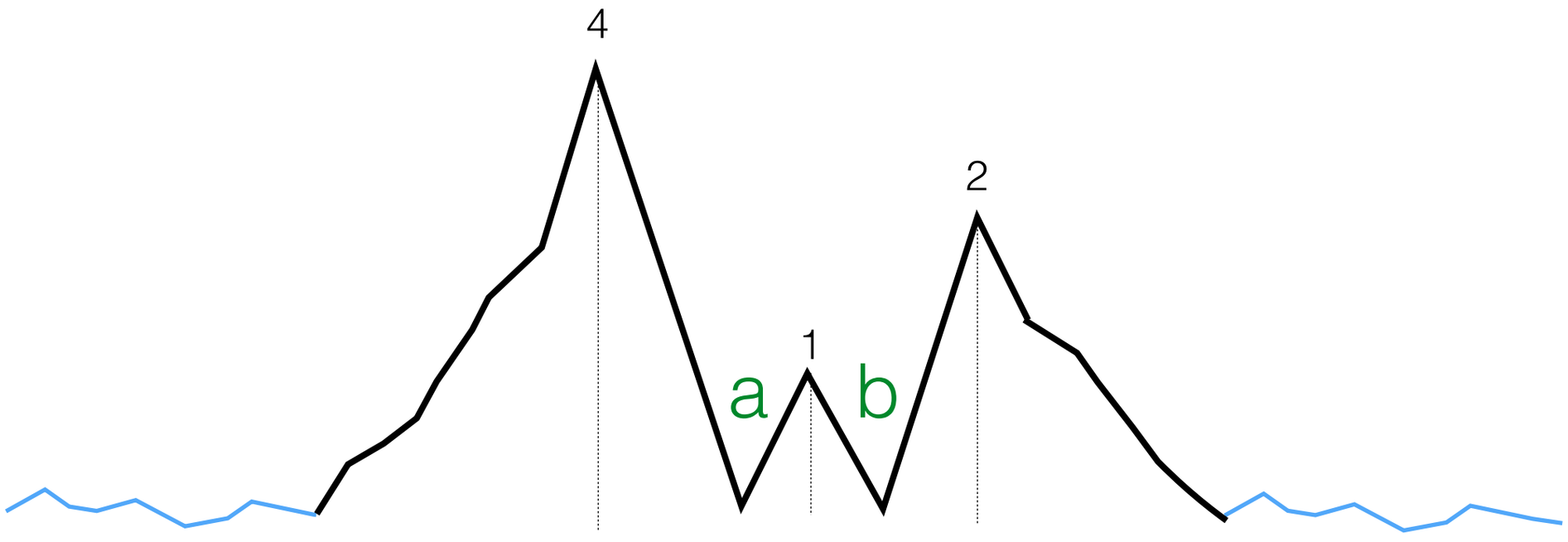


After flooding we'll get:

sinks: a, b

raise(a) = 2

raise(b) = 2



After flooding we'll get:

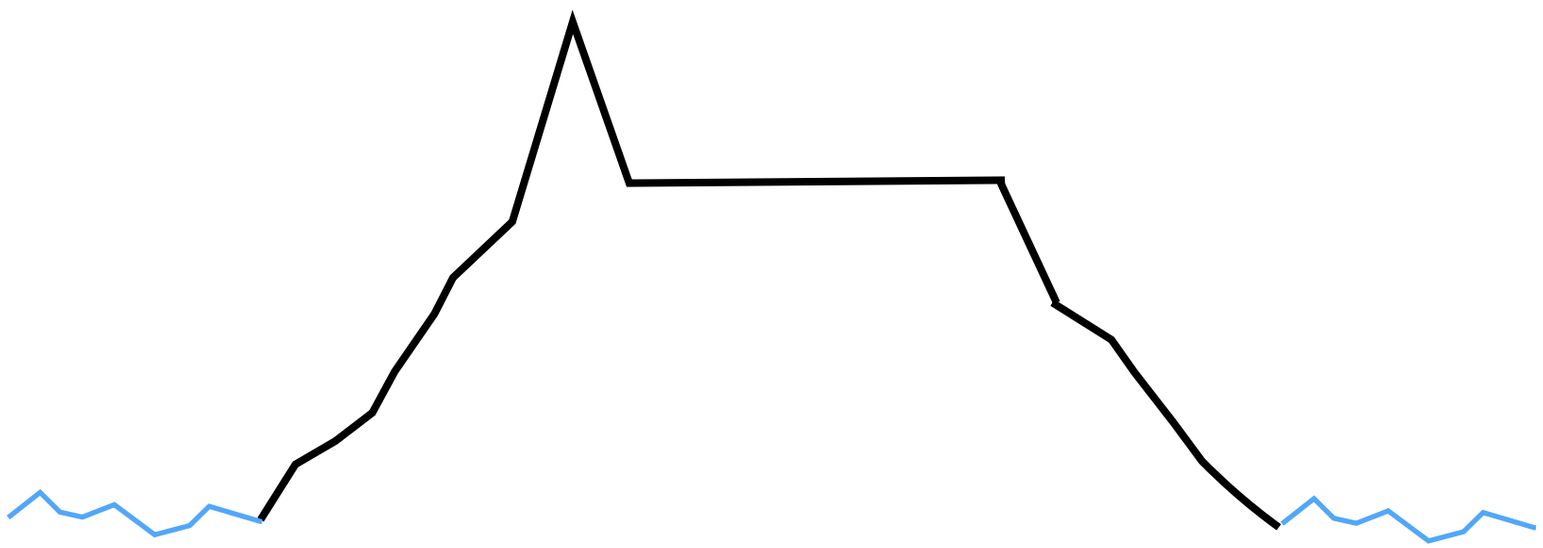
sinks: a, b

raise(a) = 2

raise(b) = 2

For each point (i,j) in the terrain: set elevation(i,j) = max (elev(i,j), raise (basin(i,j))

flooded terrain

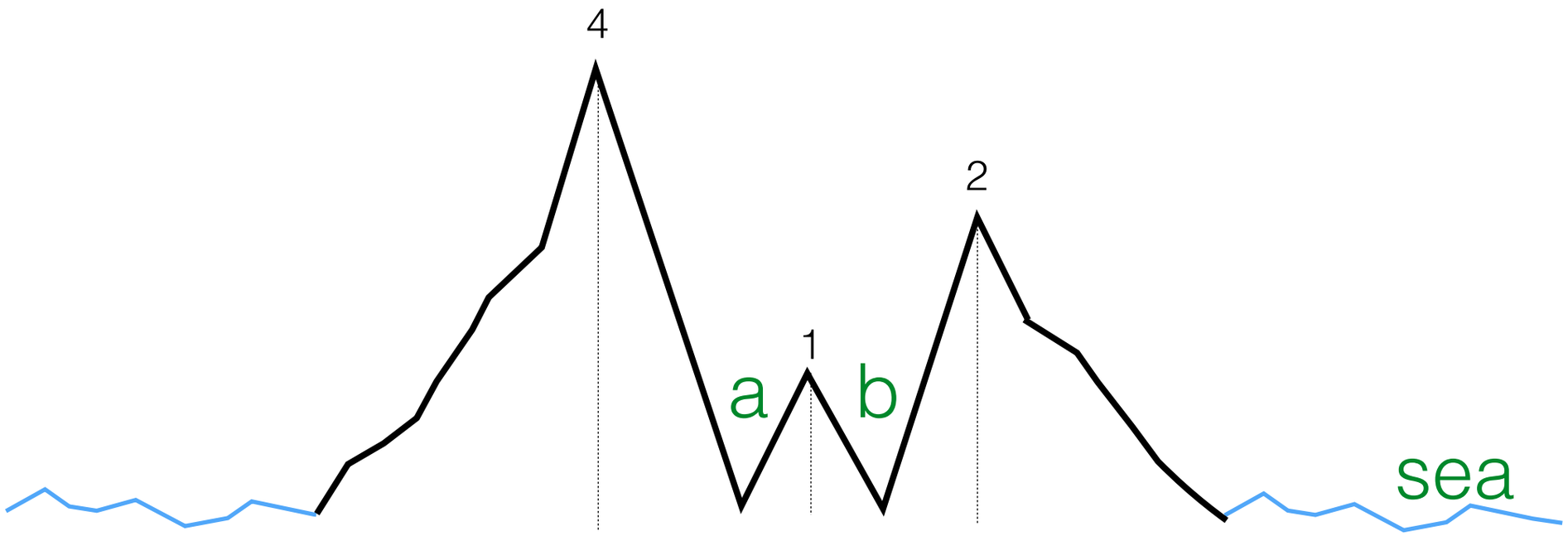


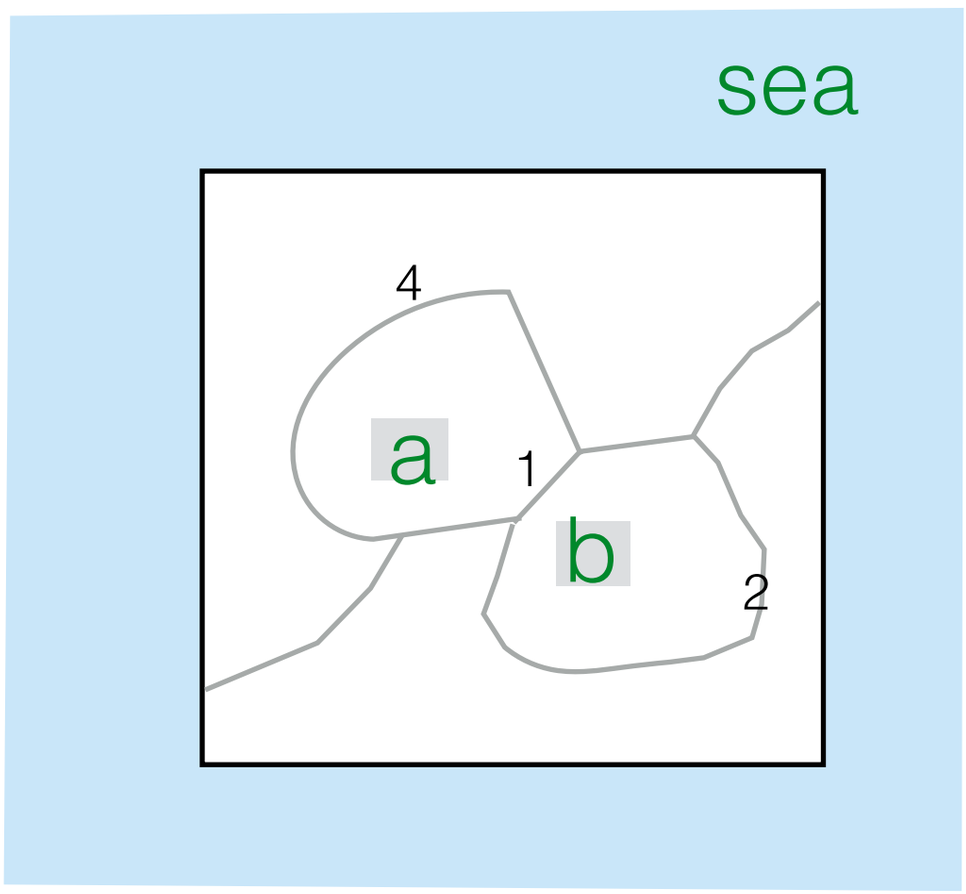
Flooding the sinks

1. Find the sinks in the terrain, including the ocean.
2. For each sink, compute its basin.
3. Find the boundaries between sink-basins.
4. Mark each edge between two sink-basins with the elevation of the lowest point along their boundary.
5. Label the ocean as DONE.
6. Sort the edges in increasing order of height
7. For next edge (s,t) at height h
    - If none of s and t are DONE, raise them to height h
    - If one of them is DONE, then the other becomes DONE and it is raised to h
    - merge s,t and update graph

Since every flow path leads to a sink or the ocean, this partitions the terrain into sink-basins.

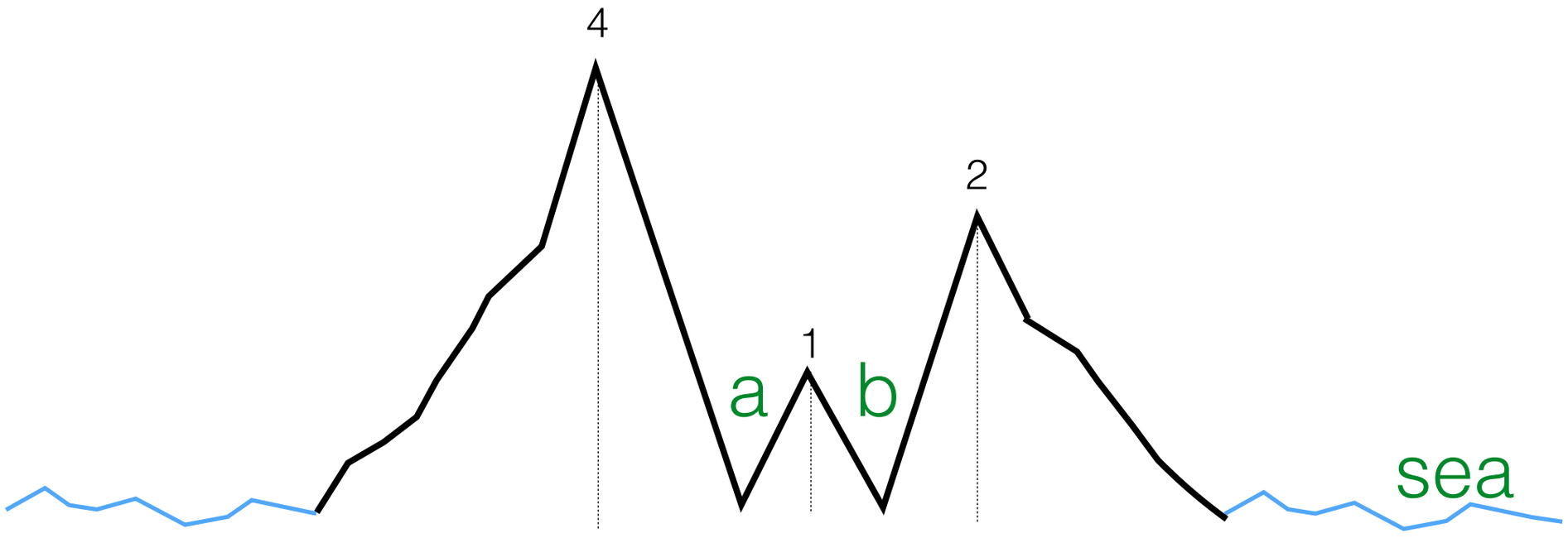
This represents the height at which s and t will merge

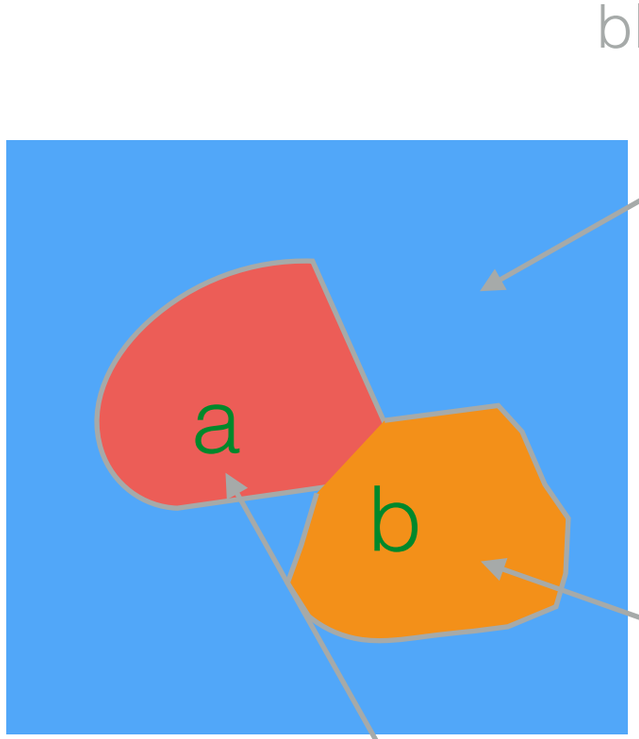




sinks: a, b, sea

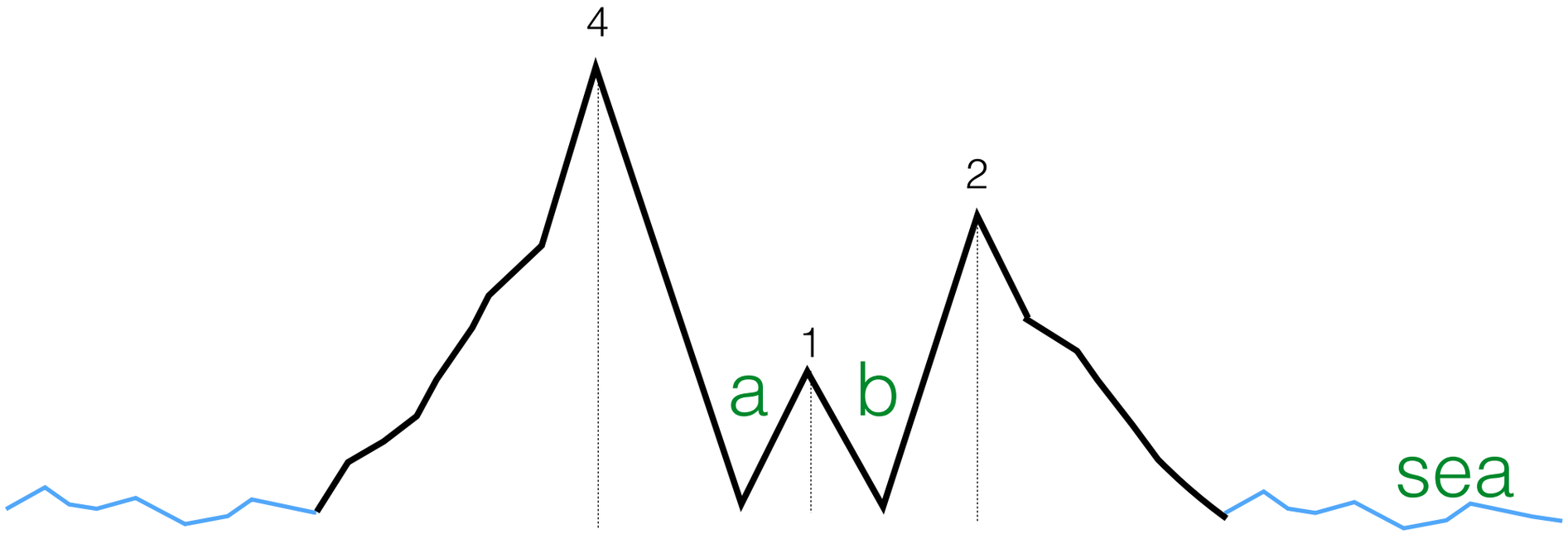
compute watershed(a), watershed(b), watershed(c)

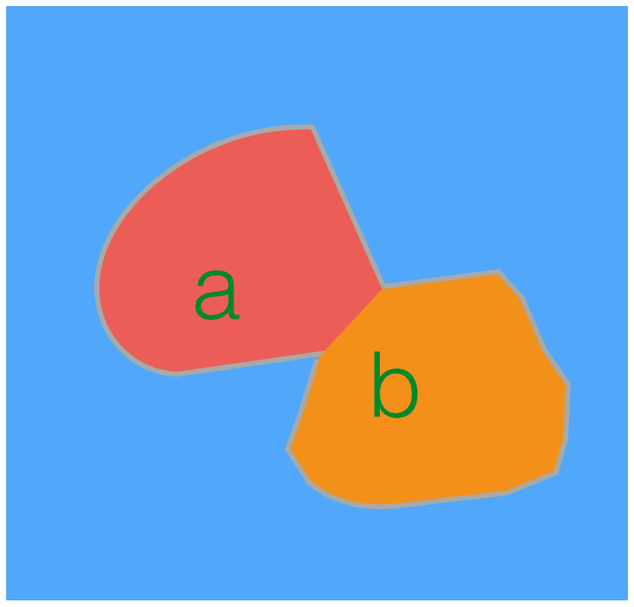




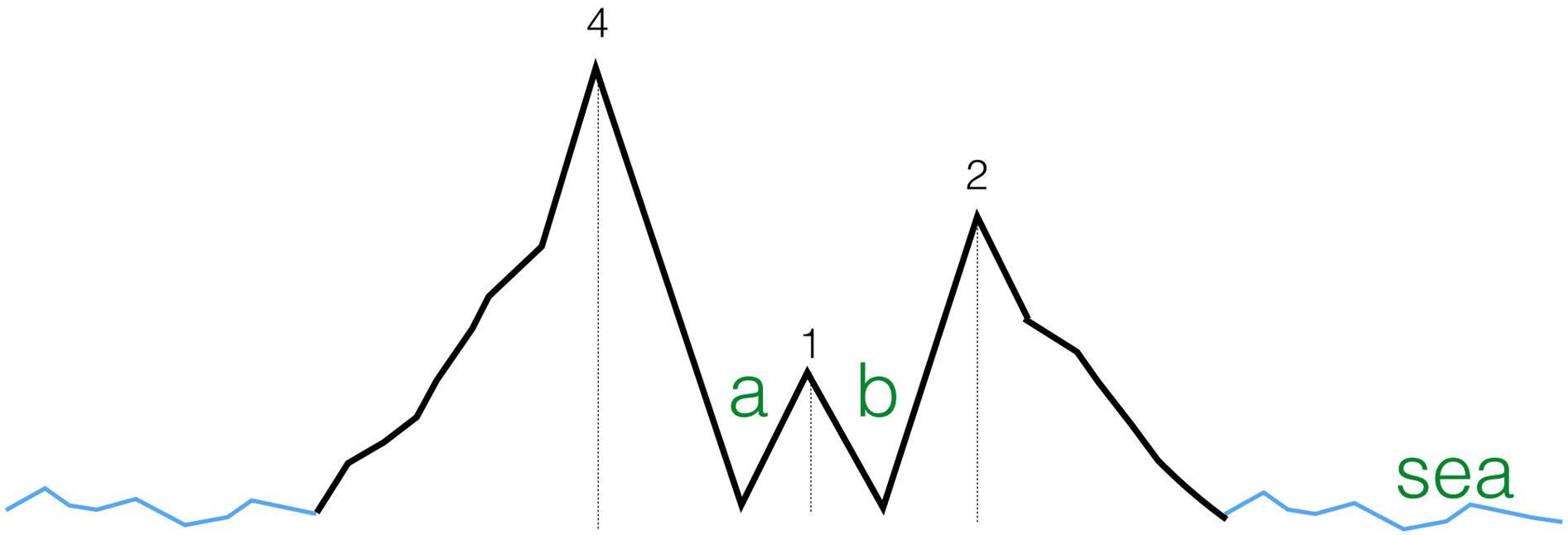
sinks: a, b, sea

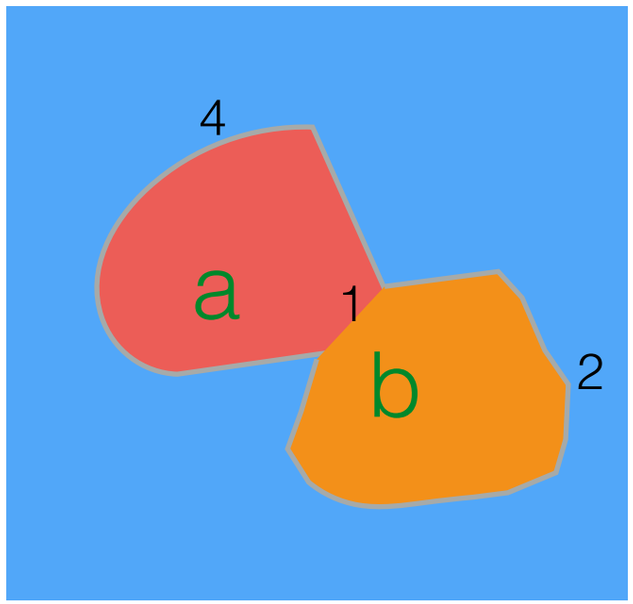
compute watershed(a), watershed(b), watershed(c)





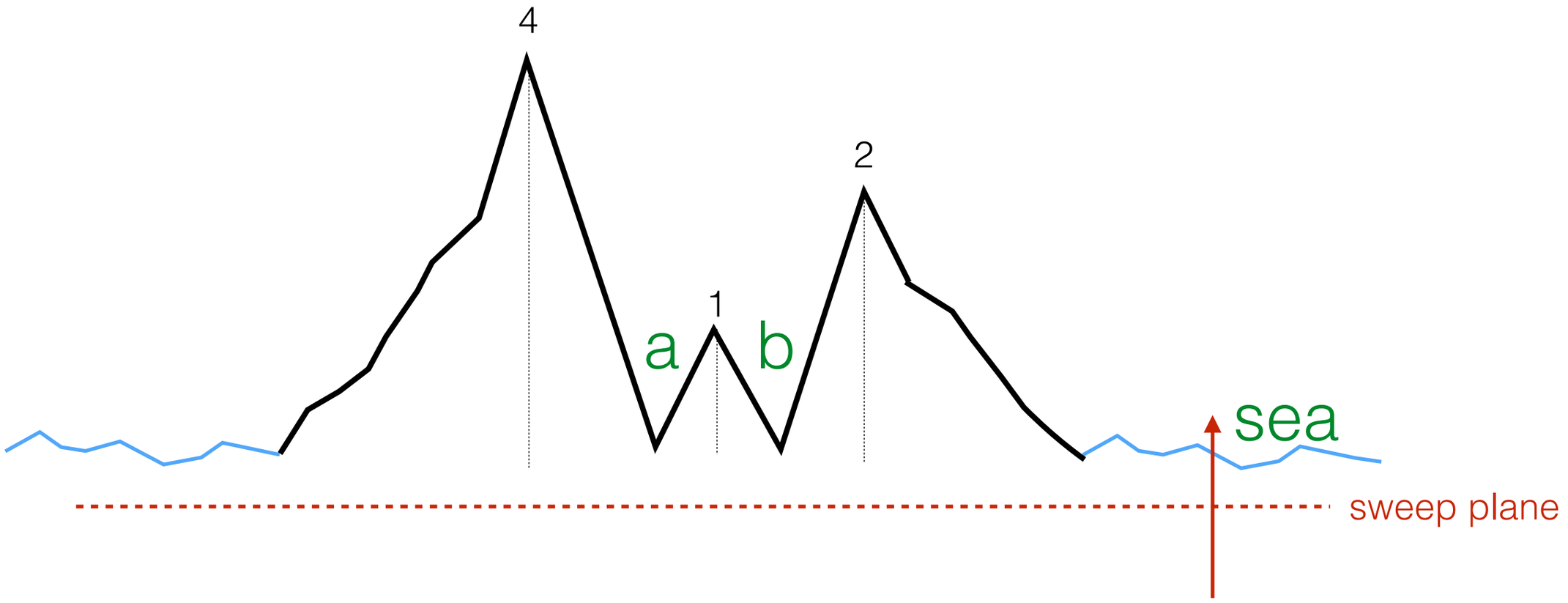
Find boundaries between sink-watersheds

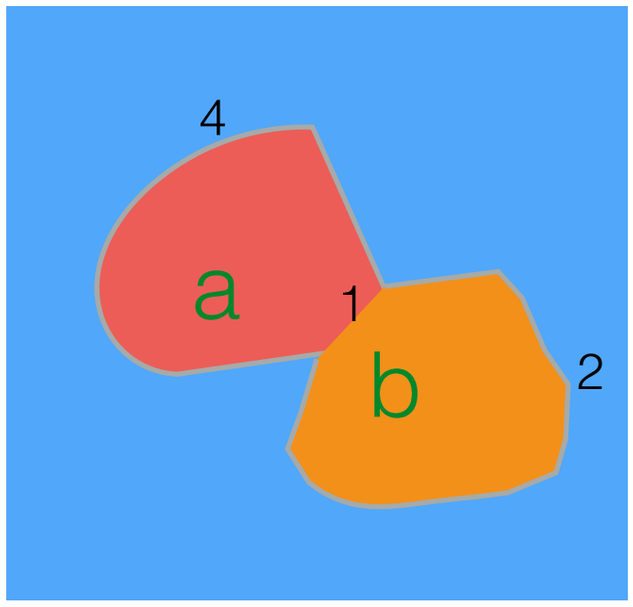


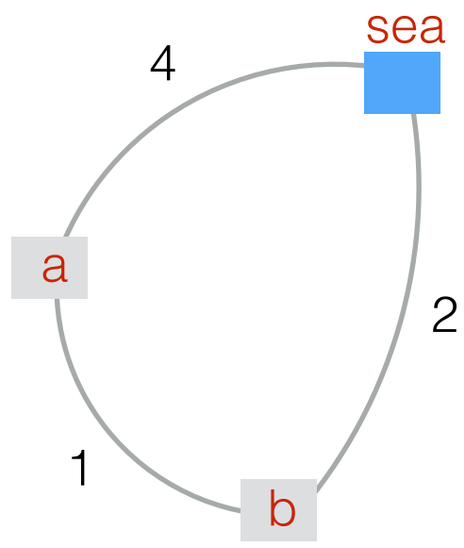


Find boundaries between sink-watersheds

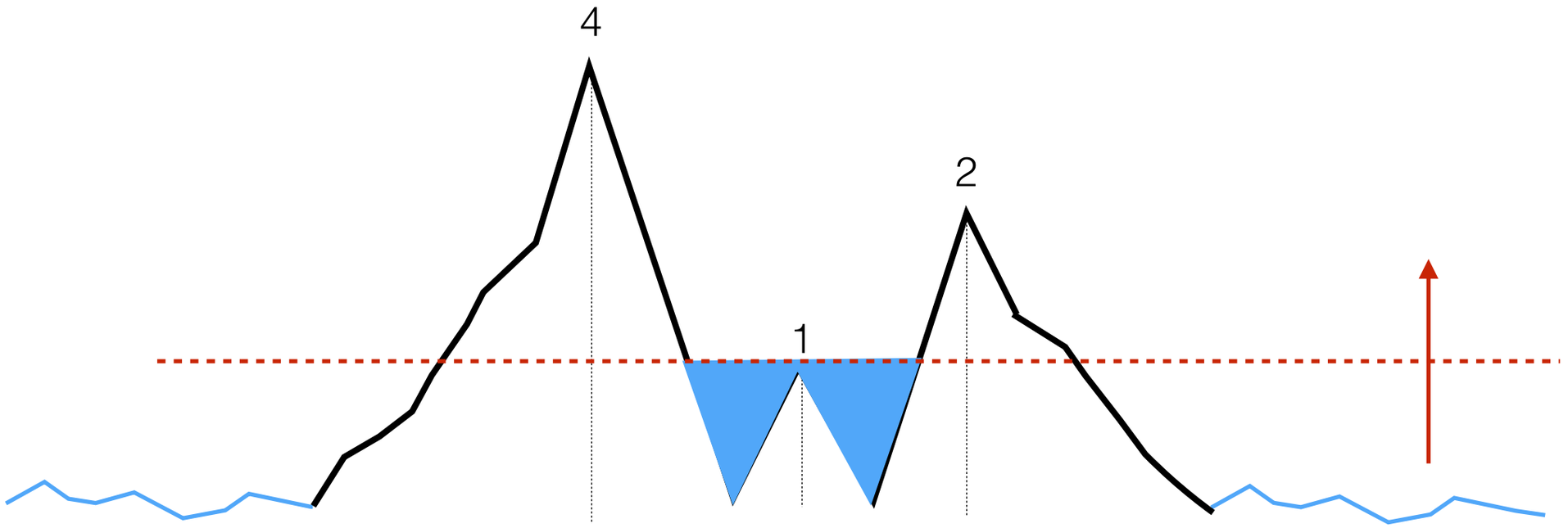
Each boundary: find lowest point along boundary

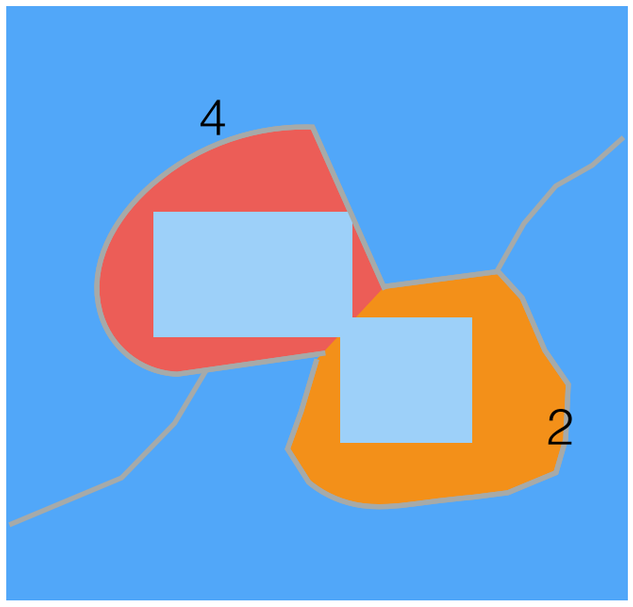


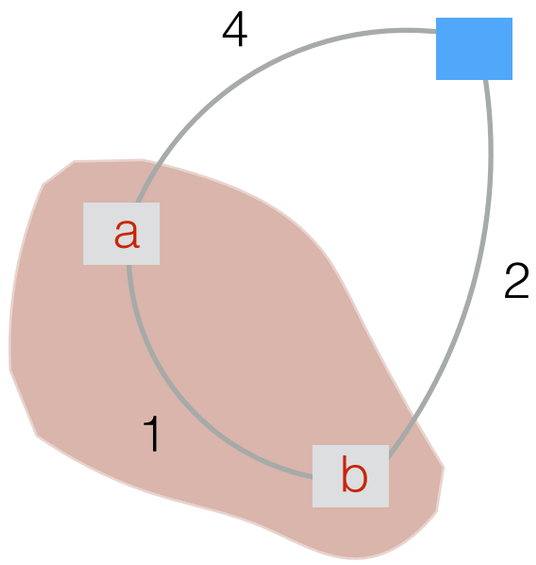




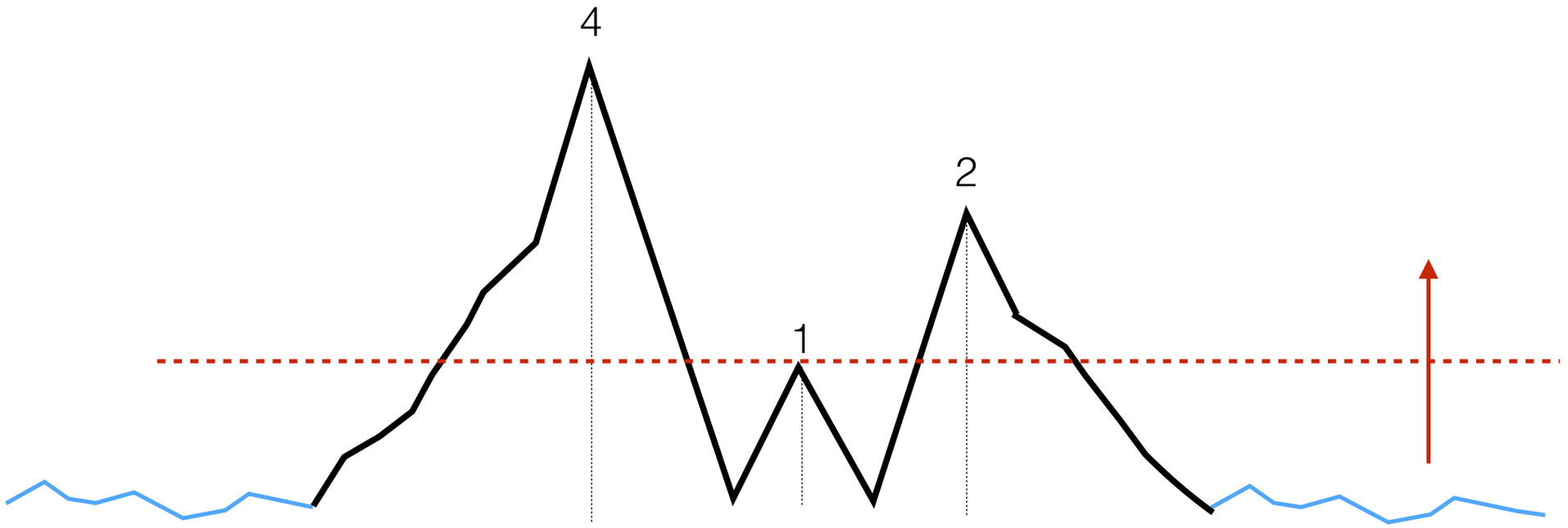
sink-watershed graph

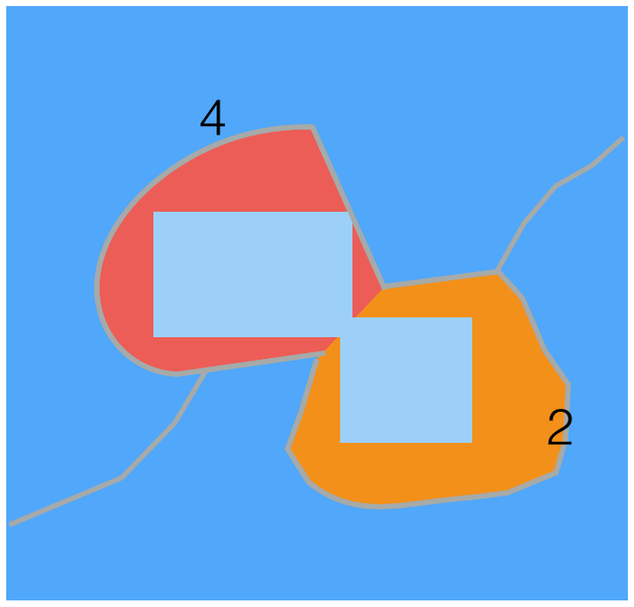


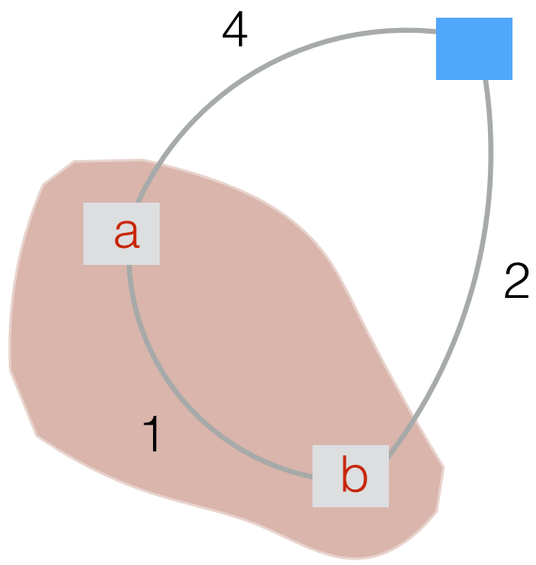




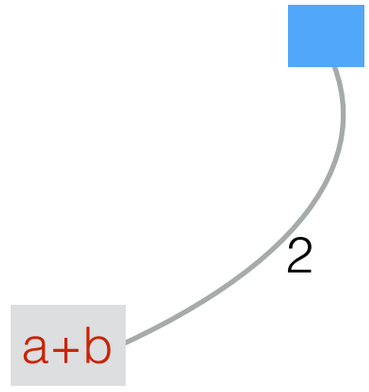
edge 1:
raise(a) = 1
raise(b) = 1
join (a,b)

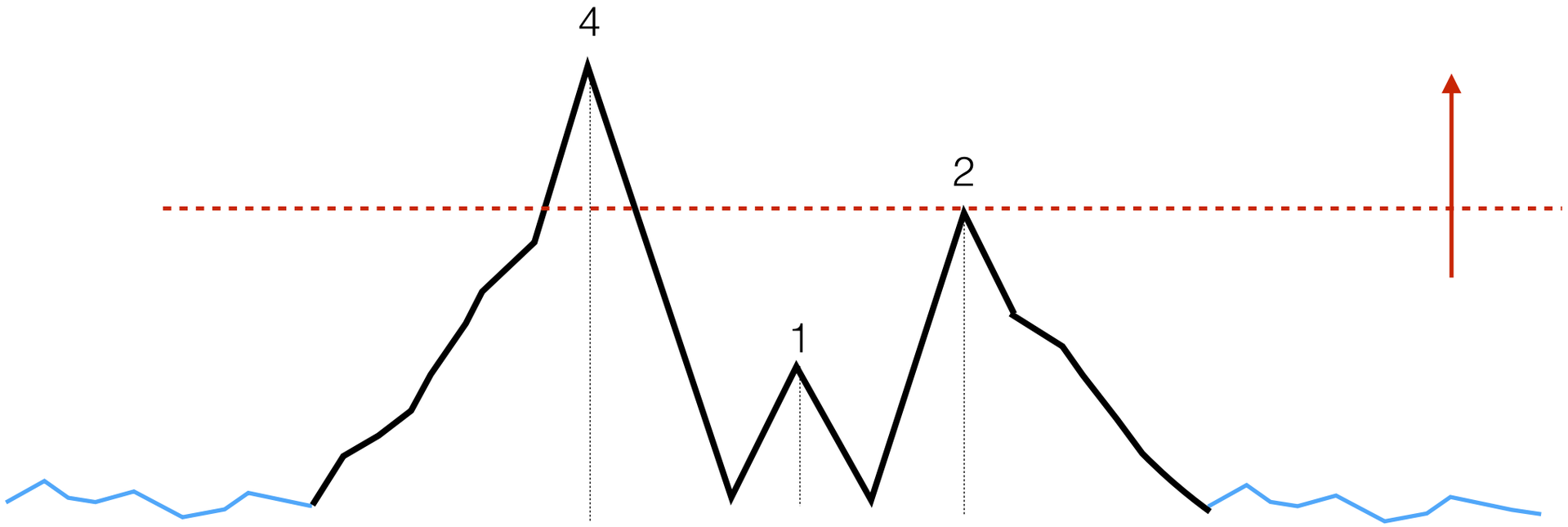


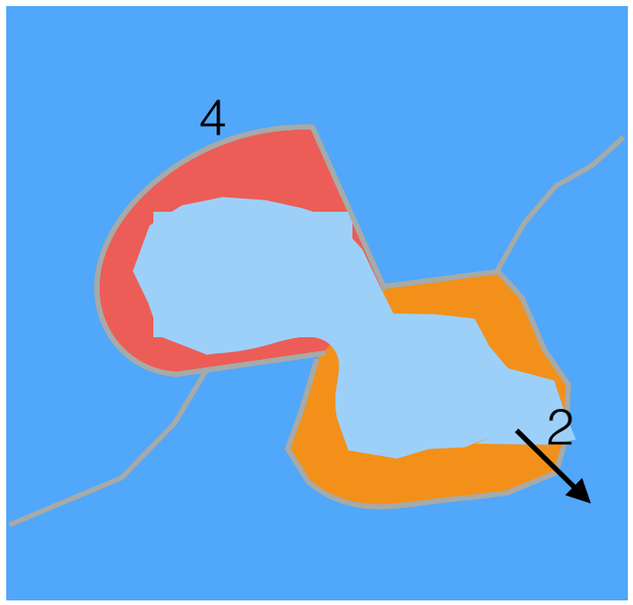


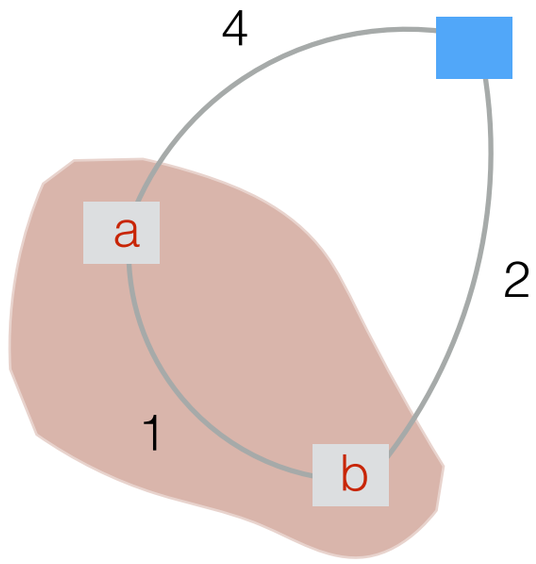


edge 1:
raise(a) = 1
raise(b) = 1
join (a,b)

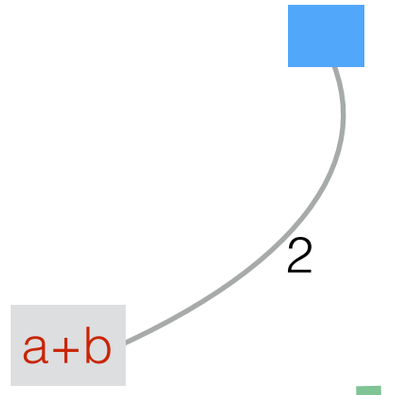








edge 1:
raise(a) = 1
raise(b) = 1
join (a,b)



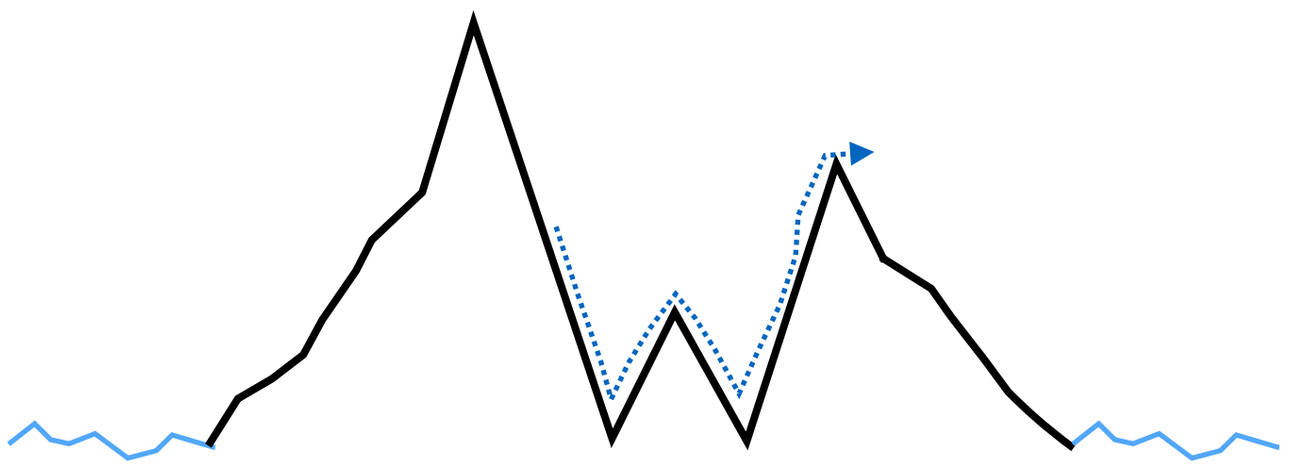
edge 2:
raise(a,b) = 2
a ,b DONE

Flooding the sinks

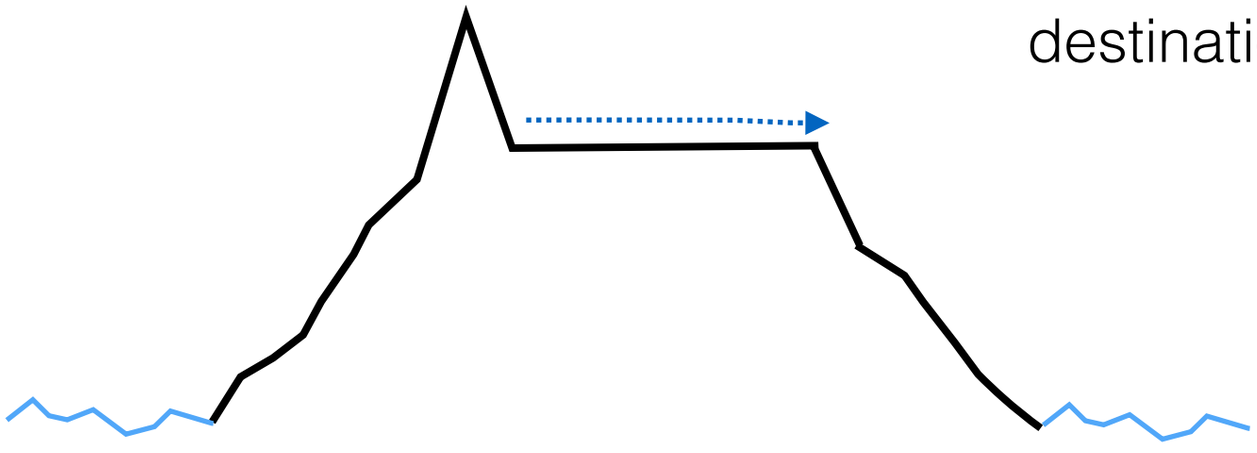
- Find the sinks in the terrain, including the ocean.
- For each sink, compute its basin.
- Find the boundaries between sink-basins.
- Mark each edge between two sink-basins with the elevation of the lowest point along their boundary.
- Label the ocean as DONE.
- Sort the edges in increasing order of height=
- For next edge (s,t) at height h
    - if s,t in the same set, ignore
    - If none of s and t are DONE, raise them to height h
    - If one of them is DONE, then the other becomes DONE and it is raised to h
    - union s and t
- Create the flooded terrain

Goal: Compute flooded terrain

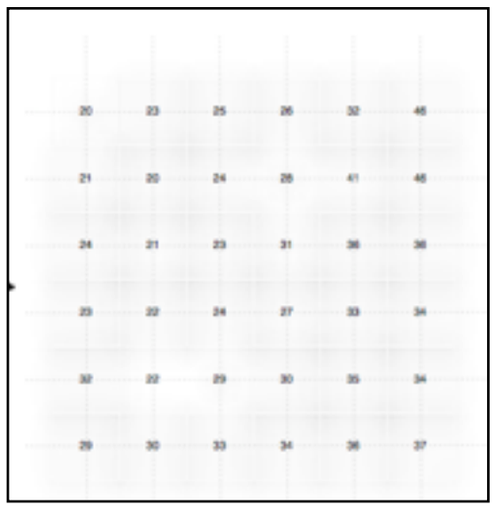
original terrain



flooded terrain

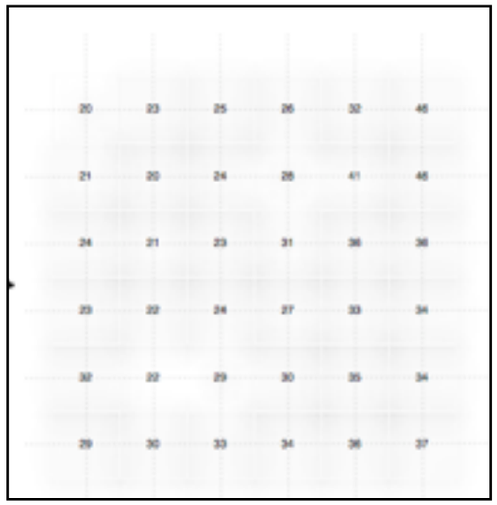


- Compute FD on the flooded terrain
- Mapped on the original terrain, FD correspond to the **lowest** path to the destination (the ocean)



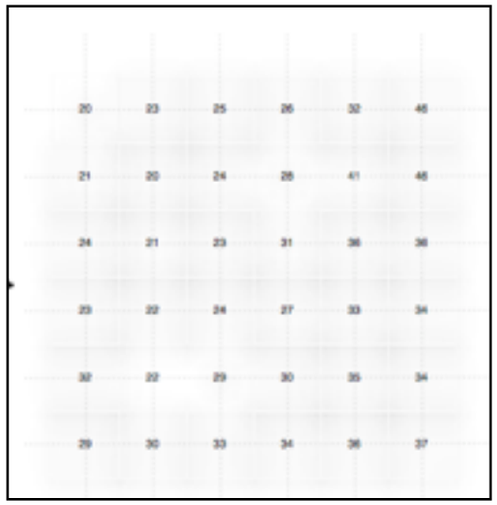
elevation grid

FD first round



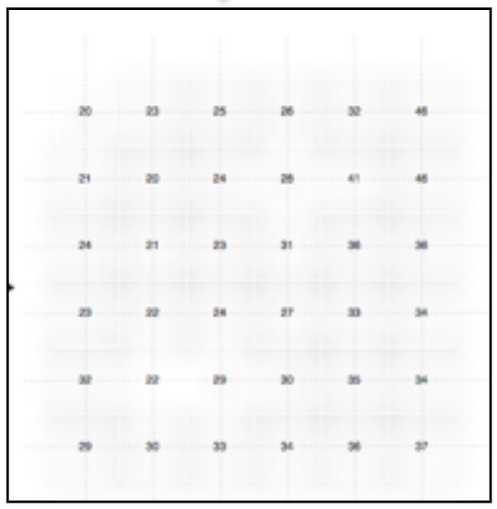
FD grid
(except plateaus and sinks)

handle plateaus



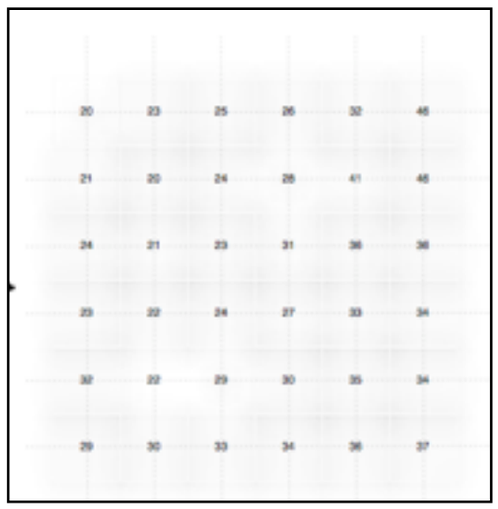
FD grid
(except on sinks)

identify sinks
and
flood sinks

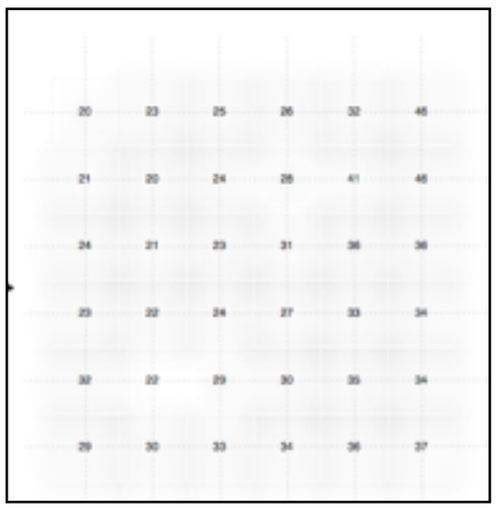


flooded elevation grid

identify
plateaus
assign FD on
plateaus



FD of flooded elevation

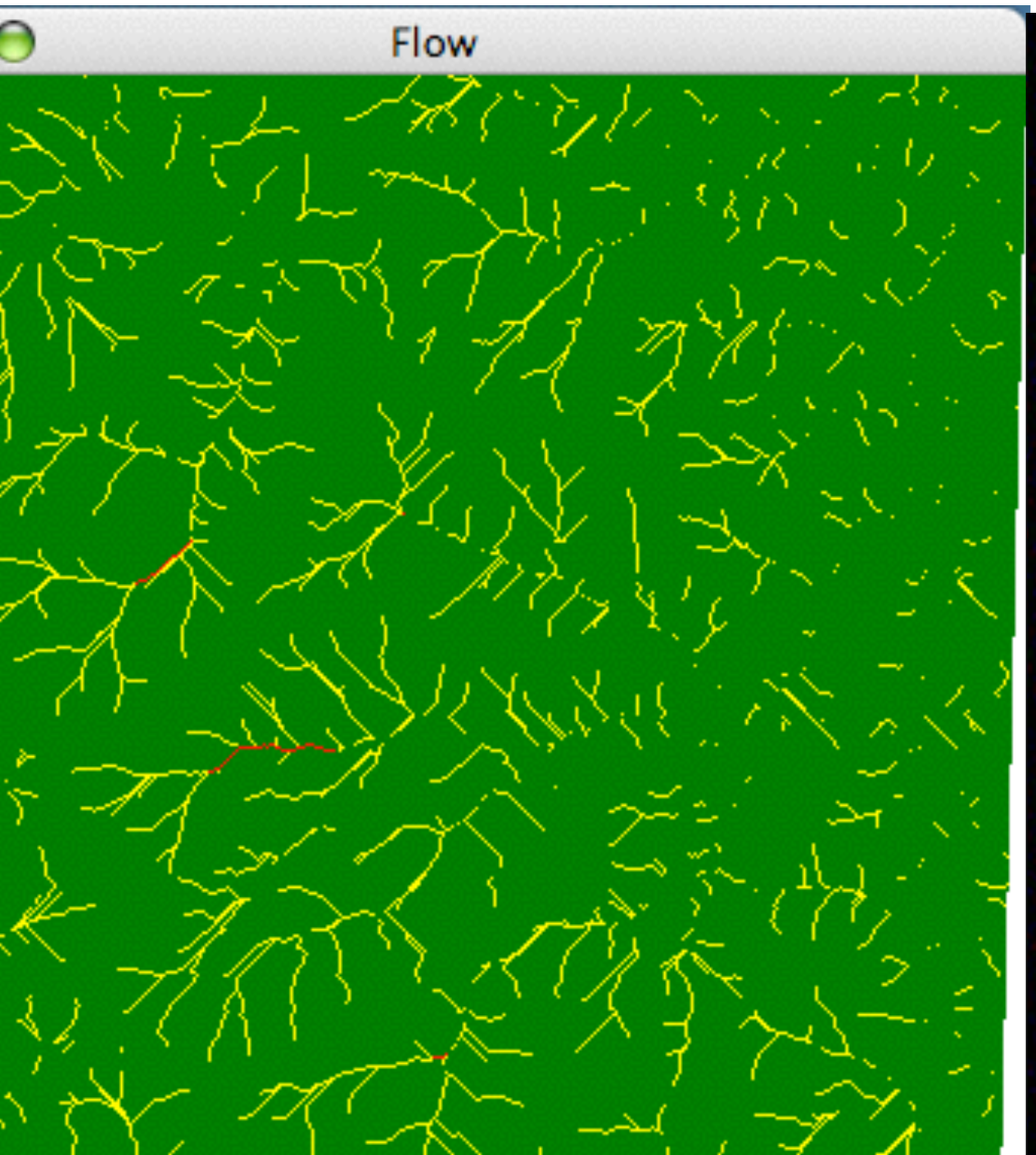


FA

It's worth it!

- Flooding creates more realistic, connected river networks.

FA: no FD on sinks



FA with flooding