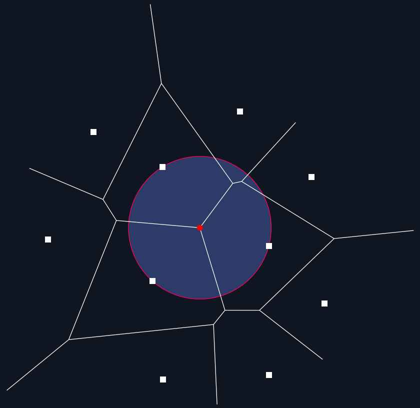
Dynamic Voronoi Diagrams

B. Zolotov, Young Researchers Seminar

Voronoi Diagrams



Voronoi Diagrams

- Every point belongs to the region of the closest *site*,
- Have clear applications everywhere,
- Can be generalized for other metrics.

Explicit Dynamic Voronoi Diagrams

There can be $\Omega(n)$ changes in coordinates per addition of a site, so time better than $O(n)$ is impossible.

Implicit Dynamic Voronoi Diagrams

($=$ Nearest neighbor query)

- Kaplan, Mulzer, Roditty, Seiferth, Sharir 2020
- Chan, 2010

Combinatorial Dynamic Voronoi Diagrams

Maintain the graph of the Voronoi diagram.

Maybe there is a sublinear algorithm for inserting a site; the coordinates of all the vertices can be output in $O(n)$.

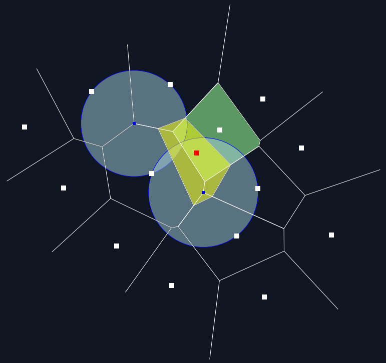
Flarb

When inserting a new site into a Voronoi diagram, you basically throw out everything inside new site's cell and replace it with a cycle. This is called a flarb.

The point is, you do not need to change every cell to implement a flarb.

Combinatorial changes

Only those cells need combinatorial changes whose Voronoi circles enclose the new site.



Chan’s polylog structure

Given a set of points in $\mathbb{R}^3$, find extreme of them in given direction: polylog time for addition and queries.

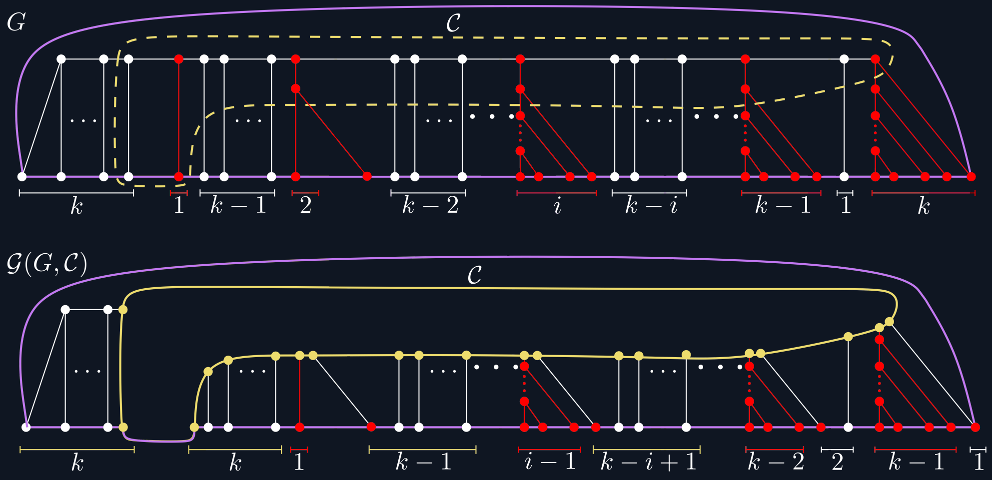
- Smallest enclosing circle
- Width of a set; convex layers
- *Report a circle containing q*

Potential function

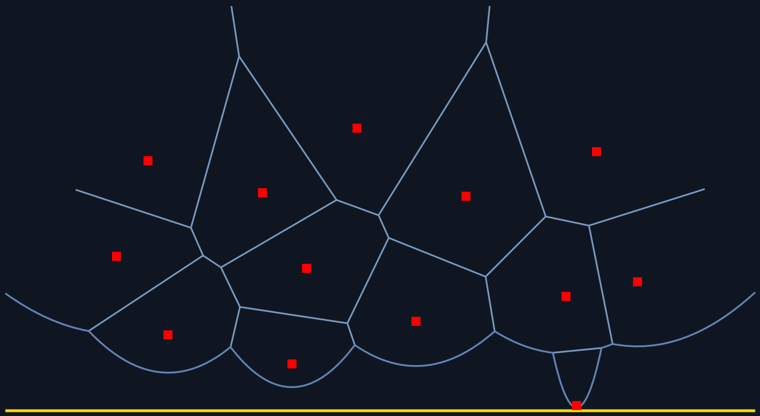
$$\Phi = \lambda \cdot \sum_{i=1}^{n} \min\left\{\mathrm{size}(f_i), \sqrt{n}\right\}$$

When a flarb is applied, all cells that undergo combinatorial changes reduce in size.

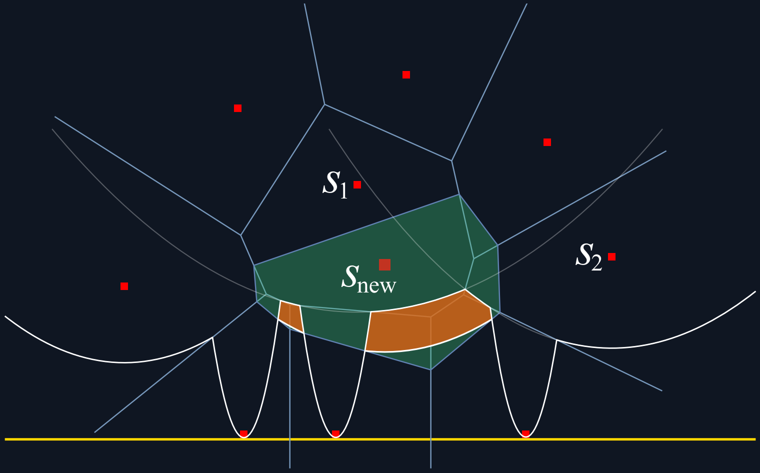
$\sqrt{n}$ lower bound



VD for several points and a line

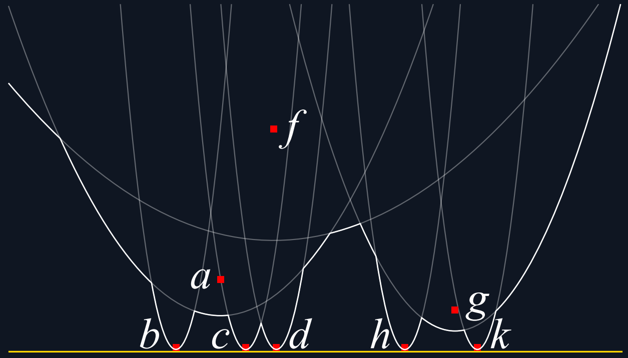


Relabel operation



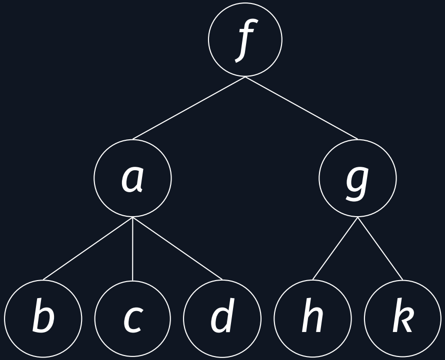
Beach line and D—S sequences

This configuration of parabolas corresponds to the sequence *fabacdafghgkgf*.

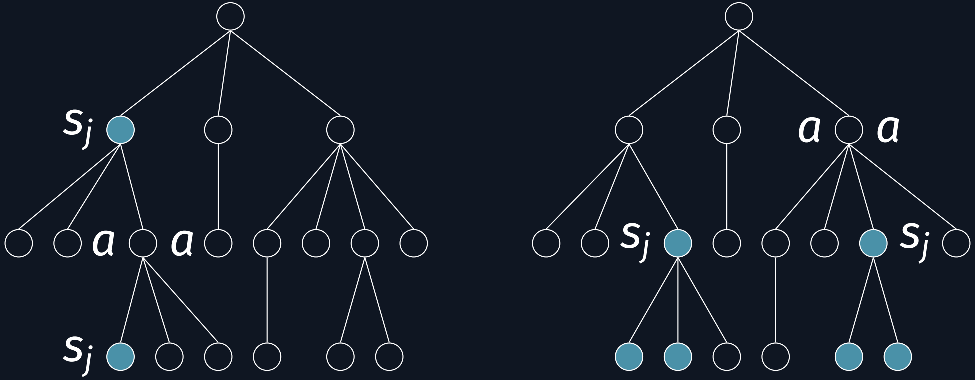


D—S sequences and trees

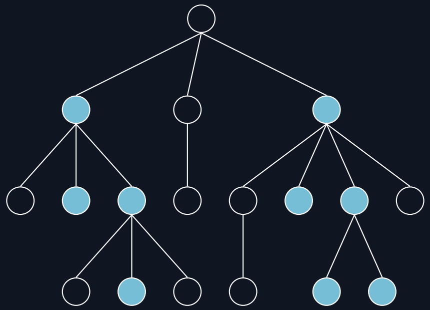
fabacdafghgkgf



The order of relabels



Possible set of relabels



Nodes that are relabelled form a union of subtrees, whose roots are children of a single node R.

Relabels and sizes of nodes

When relabels are applied, only two nodes of the tree can increase in size: the new one and R.

Inserts the length

An *insert* is a relabel that does not reduce the size of the node it is applied to.

- When an insert is applied to a sequence, the length of the sequence increases by 1.
- When a relabel that is not an insert is applied to a sequence, the length of the sequence may decrease, but by at most 2.

Potential function

$$\Phi = 3 \cdot \sum_{i=1}^{n} \min\left\{\mathrm{size}(s_i), 2\sqrt{n}\right\} - \mathrm{length}(S).$$

This accounts for the changes in the sizes of nodes and in the length of the sequence.

Further research

It suspiciously seems like you can only have significantly less than $\sqrt{n}$ relabels per letter...