Reif's algorithm \( \min (s,t) \)-cuts in planar graphs in \( O(\log n) \) time

Compute median shortest path in \( O(1) \) time
+ Recurse on both sides

---

**Monge heap**

- **Monge** \( \Rightarrow \) row minima are monotone (further right, lower rows)
- \( O(\log k) \) amortized time
- \( O(k\log k) \) time

**Input:** Monge array \( D \in \mathbb{R}^{k \times k} \)

- Hidden vector \( c \in \mathbb{R}^k \)

**Goal:** Search array \( M \)

- \( M[i,j] = D[i,j] + c[j] \)
  - For min elements in every row

- **Reveal** \((i,x)\) - set \( c[j]=x \) "revealing" column \( j \) of \( M \)

- **Find Min** - return the smallest visible entry in \( M \)
  - \( \sqrt{k} \) must be \( \min \) element (not just visible) in its row of \( M \)

- **Hide** \((i)\) - hide \( i \)th row of \( M \)
  - \( \sqrt{k} \) must contain \( \min \) element
Live intervals \((i_j, imin, imax)\) \(
\rightarrow\)
priority queue
indexed by min element

Range minimum query structure for each column \(j\)
Given \(imin\) and \(imax\) find \(\min \{B_i[j] \mid imin \leq i \leq imax\}\)

```
Static
Balanced binary tree \(k\) leaves
left stores \(D[i, j]\)
internal node store \(\min\) of children
```

```
Answer query in \(O(\log k)\) time.
```

---

FRL-Dijkstra

Given planar map \(E\) with \(v\) edges

Build nice \(r\)-division \(\rightarrow\) every piece
dense distance graph \(O(\log^2 n)\) bd verts \(O(1)\) holes

Run Dijkstra in DDG, but use Monge heaps
instead of std-pr-queue.

\(O(n \log^2 r)\) preprocessing time

Compute shortest path between two boundary nodes
of \(r\)-division in \(O\left(\frac{n}{r} \log^2 r + \log n\right)\) time

\(r = \log^6 n\)

\(T(n) = 2T\left(\frac{n}{r}\right) + O\left(\frac{n}{\log n}\right) = O(n \log \log n)\)
An image containing various mathematical and geometric diagrams. The text includes symbols and mathematical expressions, but the specific content is not legible due to the handwriting and drawing style. The text appears to discuss some form of algorithm or mathematical problem, possibly related to computational geometry or graph theory, with references to 'Monge arrays' and 'total area' and 'total perimeter'. The diagrams show interconnected nodes and paths, indicating a network or graph structure.
FR-Dijkstra:

Maintain a Monge heap for every Monge array associated with every piece of division.

Maintain a piece heap for each piece containing min elements in every Monge heap in that piece.

Maintain a global heap containing min elements of piece heaps.

Init: Empty everything

Reveal(s, 0) in every Monge heap

Hide(s) relevant to s

Repeat

Extract Min ≤ v

Reveal (v, dist(v)) every Monge heap relevant to v

Hide (v)

Total time to manage one k × k Monge heap = O(k log k) per perimeter

Total time to manage all Monge heaps =

\[O\left(\frac{n}{\sqrt{r}}\right) \times O\left(\sqrt{r} \log r\right) \times O\left(\log r\right) = O\left(\frac{n}{\sqrt{r}} \log^2 r\right)\]

Total time to manage all piece heaps

\[O\left(\frac{m}{\sqrt{r}}\right) \times O\left(\sqrt{r} \log r\right) \times O\left(\log r\right) \times O\left(\log r\right) = O\left(\frac{m}{\sqrt{r}} \log^2 r\right)\]

Total time for global heap = \(O\left(\frac{n}{\sqrt{r}} \log r\right)\)
FR-Dijkstra: $O(n \log r)$ prep time

- build nice $r$-division
- dense distance graph
- init Monge heaps (range min trees)

Query: Compute shortest path (distance) between two nodes in DDG in

$$O\left(\frac{n}{r} (\log^2 r + \log n)\right)$$ time

$$r = O(\log n)$$

$O(n \log \log n)$ prep $O\left(\frac{n}{r} \log \log n\right)$ query time

$\min (s, t)$ - cut $\rightarrow$ Monge matching

\[
\begin{align*}
\text{For all } i & \\
& \text{shortest path distance from } s_i \text{ to } t_i
\end{align*}
\]

Italiano Nussbaum Sankowski Wulff-Nilsen 2011

- Prep for FR-Dijkstra $O(n \log r) = O(n \log \log n)$
- Coarse divide-conquer phase:
  $O(k \log n)$ equally spaced paths following Zeit
  using FRD
- Fine divide-conquer phase:
  Run Zeit's algo in each slab $O(n \log \log n)$

\[
T(n, k) = T\left(n_2, \frac{k}{2}\right) + T\left(n_2, \frac{k}{2}\right) + O\left(\frac{n}{r} (\log^2 r + \log n)\right)
= O\left(\frac{n}{r} \log^3 n\right) = o(n) \quad (r = \log^k n)
\]

$\Rightarrow O(n \log \log n)$ time
Coarse → Fine: We need shortest paths in $\mathbb{Z}$ not just in DDG.

DDG was computed using MSSP algo.

We can extract $\text{pred}(v)$ in any $T_c$ in $O(\log \log \deg(v)) = O(\log \log n)$.

But we need $\frac{k}{\log n}$ paths.

They might overlap.

Union = tree forest $\Rightarrow O(n \log \log n)$

In coarse phase, need to slice dense distance cliques.