# Dynamic Graph Connectivity: Modern Algorithms

- Kapron-King-Mountjoy (KKM) Algo by graph sketching (2013)
- Nanongkai-Saranurak Algo using expanders (2017)

## Dynamic Graph Connectivity (undirected)

Given an initial graph and a sequence of updates Insert(e) and Delete(e), determine after each update whether the graph is connected.

Naive approaches:

① Run DFS/BFS on each query: update time $O(1)$, query time $O(m)$

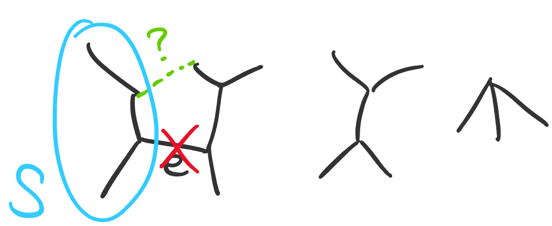
② Run DFS/BFS on each update: update time $O(m)$, query time $O(1)$

KKM Algo: maintain a maximal spanning forest.

On Query: Connected $\Longleftrightarrow$ forest is a spanning tree

On Insert(e): If e connects two trees in the forest, then add to forest. Else, do nothing.

On Delete(e): If $e \notin$ forest, do nothing.
Else $e \in$ forest. Remove e and <u>find a replacement edge</u> if it exists




Let $S$ be one component after removing e.
Suffices to find an edge out of $S$ if it exists.

given a tree on vertices $S$ in the forest, find an edge out if it exists.

Task: given a tree on vertices $S$ in the forest, find an edge out if it exists.

Arbitrarily direct the edges. For each vertex $v \in V$, consider

$$x_v \in \mathbb{R}^E: \quad x_v = \sum_{e \in \partial^+ v} u_e - \sum_{e \in \partial^- v} u_e = [0 \cdots 0 \underset{e \in \partial^+ v}{+1} 0 \cdots 0 \underset{e \in \partial^- v}{-1} \cdots]$$

$u_e$ = unit vector for $e$

$\partial^+ v$ = edges directed out of $v$

$\partial^- v$ = " " into $v$

Now consider $x_S = \sum_{v \in S} x_v \in \mathbb{R}^E$.

Claim: $x_S = \sum_{e \in \partial^+ S} u_e - \sum_{e \in \partial^- S} u_e$. In particular, $x_S \neq 0 \iff$ exists edge into/out of $S$.

Proof: any edge $u \xrightarrow[e]{} v$ internal to $S$ $(u, v \in S)$ is cancelled out:

$x_u$ has $+u_e$ and $x_v$ has $-u_e$.

Slow algo: after deleting edge $e$ from forest, let $S$ be one side.

If $x_S = \sum_{v \in S} x_v \neq 0$, then find a nonzero coordinate $e$

and add $e$ as replacement edge.

Speedup: dimension reduction!

Re-define $u_e \in \{0,1\}^{2 \log n}$ as $\left[\underbrace{\overbrace{01011 \cdots 0}^{\log_2 n}}_{\text{bit ID of } u} \; \underbrace{\overbrace{101 \cdots 0}^{\log_2 n}}_{\text{bit ID of } v}\right]$ "unique fingerprint"

$u \xrightarrow[e]{} v$

Still have $x_S = \sum_{e \in \partial^+ S} u_e - \sum_{e \in \partial^- S} u_e$. Still $x_S \neq 0 \Rightarrow$ exists edge into/out of $S$,

but $\not\Leftarrow$ due to possible cancellations.

But if exists <u>single</u> edge into/out of $S$, then $x_S = \pm u_e$, and

we can recover edge $e$ by reading off the bit IDs.

If there exists more? Subsample!

Let $E_i \subseteq E$ be a subsample where each edge $e$ sampled w.p. $1/2^i$.

Build data structure separately for each $E_i$.

If $|\partial S| = k$, then for $i = \lfloor \log_2 k \rfloor$, sampled each $e \in \partial S$ into $E_i$ w.p. $1/2^i \approx 1/k$. So with constant probability, sampled exactly one, and data structure recovers the sampled edge.

How to boost probability? Maintain $O(\log n)$ samples $E_i$ for each $i$.

With prob $\geq 1 - \frac{1}{\text{poly}(n)}$, at least one sample succeeds.

Running time?

Query: calculate $\sum_{v \in S} x_v$ for each of $O(\log^2 n)$ samples $E_i$.

Since $S$ induces a tree, can be done using dynamic trees.

Total $\text{poly}\log(n)$ time.

Update: on edge $e = (u,v)$, only need to update $x_u, x_v$ in each sample $E_i$. Update dynamic tree data structure.

Total $\text{poly}\log(n)$ time.

---

Dynamic Connectivity using Expanders

Def (Expander): A graph is a $\phi$-expander if

$$\forall S \subseteq V, \quad \frac{|E(S, \bar{S})|}{\min\{\text{vol}(S), \text{vol}(\bar{S})\}} \geq \phi .$$

$\text{vol}(S) = \sum_{v \in S} \deg(v)$

volume = sum of degrees

Suppose the initial graph is a $\phi$-expander.

Lemma: Suppose the initial graph is a $\phi$-expander.
For any inputs $D \subseteq E$ and $s,t \in V$, we can check whether $s,t$ are connected in $G \setminus D$ in $O(\frac{|D|}{\phi})$ time.

Proof: Suppose $s,t$ disconnected in $G \setminus D$. Let $S \subseteq V$: $s \in S$, $t \notin S$.
Since $G$ is a $\phi$-expander, $|E(S,\bar{S})| \geq \phi \min\{vol(S), vol(\bar{S})\}$.
All edges of $E(S,\bar{S})$ are deleted, so $|D| \geq |E(S,\bar{S})| \geq \uparrow$.
Rearranging gives $\min\{vol(S), vol(\bar{S})\} \leq \frac{|D|}{\phi}$.

The algorithm runs DFS/BFS from $s$ but terminates after reaching total volume $> \frac{|D|}{\phi}$. Repeat for $t$. Running time $O(\frac{|D|}{\phi})$.
If $s,t$ disconnected, then either $vol(S) \leq \frac{|D|}{\phi}$ (where $s \in S$)
or $vol(\bar{S}) \leq \frac{|D|}{\phi}$ (where $t \in \bar{S}$).

So DFS/BFS must find an entire connected component separating $s$ from $t$. So certify that $s,t$ disconnected.
Otherwise, $s,t$ connected.