Degree-0 persistence (a.k.a. Size Theory)

Input: \( f : X \to \mathbb{R} \)

Idea: look at (path-) connected components of excursion sets of \( f \):
- sublevel sets: \( f^{-1}(-\infty, t] \)
- superlevel sets: \( f^{-1}(t, +\infty) \)

Hyp: \( f \) is tame, i.e., every excursion set has finitely many (path-) connected components (CCs).

Def. \( F(t) := f^{-1}(-\infty, t] \) (sublevel set)

\[ \Pi_0 F(t) := \{ \text{CCs of } F(t) \}. \]

Obs. \( A \leq t, \forall C \in \Pi_0 F(s), \exists ! C' \in \Pi_0 F(t) \) s.t. \( C \cap C' \neq \emptyset \) (in fact \( C \subseteq C' \)). This is because the CCs grow with\( t \) and are pairwise disjoint.

\( \Rightarrow \) there is an induced map \( \varphi(s, t) : \Pi_0 F(s) \to \Pi_0 F(t) \) that tells where each CC of \( F(s) \) "goes" in \( F(t) \).

Examples:
Define: Given \( F \in \mathbf{CR} \) and \( C \subseteq \Pi_0 F(t) \):

- \( b(c) := \inf \left\{ s \leq t \mid c \subseteq \operatorname{Im} \gamma(s, t) \right\} \) [birth]
- \( d(c) := \sup \left\{ m > t \mid \forall s \in \gamma(t, m) \neg \left( \gamma(t, m)(c) \subseteq c \right) \right\} \) [death]

\( [b(c), d(c)] \) is the box corresponding to \( C \) in the barcode of \( f \).

Formally:

\[
\operatorname{Barcode}(f) := \left\{ [b(c), d(c)] \mid C \subseteq C_0 \right\}
\]

where

\[
C := \{ C \mid C \subseteq \Pi_0 F(t) \text{ for some } t \in \mathbf{R} \}
\]

\( \hat{C} := \{ C \mid C \subseteq \Pi_0 F(t) \text{ for some } t \in \mathbf{R} \} \) where \( C \prec C' \Rightarrow \gamma(t, m)(C) = C' \)

\[
\Pi_0 F(t) \overset{\sim}{\to} \Pi_0 F(m)
\]

and \( m \leq d(c) \) if \( C \) is still an independent \( C \)

Def. \( \operatorname{Dy} f := \{ (b(c), d(c)) \in \mathbf{R}^2 \mid C \subseteq C_0 \} \) (\( \Delta \) this is a multiset).

Thin (stability):

For any two functions \( f, g : X \to \mathbf{R} \),

\[
\| \phi (D_{\infty} f, D_{\infty} g) \|_{\infty} \leq \| f - g \|_{\infty}
\]
Computing degree-0 persistence
to record connected components

Input: graph $G = (V, E)$, map $f: V \cup E \rightarrow \mathbb{R}$

Type: graph filtration, i.e., $f(v, v) \geq \max \{ f(u), f(w) \}$

Preproc:
- Sort $V \cup E$ by increasing lexicographic order ($f$-value, then $\delta$).
- Get a sequence $(v_1, \ldots, v_m)$ of vertices/edges.
- Initialize a union-find data structure $U$.

Main loop:
- for $i = 1$ to $m$ do:
  - if $v_i$ is a vertex $v$ then:
    - create new entry $e_v := \{v\}$ in $V$
    - add $v$ to c.c. by $v$
  - else $v_i$ is an edge $(u, v)$:
    - find entries $e_u$ and $e_v$ containing
      respectively $u$ and $v$ in $U$
    - if $e_u \subset e_v$ then:  // assume $\delta$ order that
      - merge $e_v$ into $e_u$ in $U$
      - print out “$\cdot$” + $f(e_u)$ + “,” + $f((u, v))$ + “$\cdot$”

Post proc:
- for each remaining entry in $V$:
  - print out “$\cdot$” + $f(e_v)$ + “,” + $\infty$”

Running time:
- $\text{Preproc: } O(m \log n)$
- $\text{Main: } O(m \cdot \log n)$
- $\text{Post Proc: } O(m \cdot \text{union-find})$

Example:
- Lexicographic order:
  - $(b, a, \{b, a\}, f, d, c, \{c, d\}, \{b, c\}, \{c, d\}, \{a\})$
- Output:
  - $\{\{a, b\}, \{b, c\}, \{c, d\}, \{e, f\}, \{c, e\}, \{c, d, e\}, \{a, f\}\}$