**Motivation**

- Given a graph $G(V, E)$, where $V$ is the set of vertices, and $E$ the set of edges, consider the following problem: What is the largest subset of $V$ where every two vertices are pairwise nonadjacent?

**The Maximum (Cardinality/ Weight) Independent Set Problem**

- Why do we care?
  - Scheduling, Biology, Coding Theory ...
- Very easy to trick a greedy algorithm:

- The (Weighted) Maximum Independent Set, (W)MIS, problem is NP-hard on arbitrary graphs.

**Cocomparability Graphs**

- A cocomparability graph is the complement of a comparability graph (i.e. induced by a partial order).
- **Vertex Ordering Characterization:**
  - $G$ is a cocomparability graph iff $V$ admits an ordering $(V)$ where every triple $a < b < c$ with $ac \in E$, $ab \in E$ or $bc \in E$ or both. For example:

- Such ordering is called an umbrella free ordering.
- $O(m + n)$ to compute – McConnell & Spinrad [1].
- Cocomparability graphs are a superclass to:
  - Trapezoid and Permutation graphs
  - Cographs
  - Interval graphs

Solving a problem on cocomparability graphs yields a solution to all these graph classes!

**The Algorithm**

- Given a cocomparability graph $G(V, E)$:
  - Compute a valid cocomparability ordering $\sigma$.
  - Scan $\sigma$ from left to right to compute a new ordering $\tau$ of $V$, where vertices are inserted in $\tau$ in nondecreasing order of their (updated) weight.
  - Scan $\tau$ from right to left to greedily collect a maximum weight independent set.

**Proof of Correctness**

- Associate with every vertex a set $S(v)$, then at every iteration $i$:
  - For every vertex $v$, $S(v)$ is an independent set.
  - Every $S(v)$ is a maximum weighted independent set containing $v$ in $G[v_1, ..., v_i]$.
  - Let be $z$, the rightmost vertex of $\tau$, then $S(z)$ is a maximum weighted independent set in $G[v_1, ..., v_i]$.

**Future Work**

- Certify the algorithm. There exists a certifying algorithm for the unweighted case that computes a minimum clique cover of equal cardinality [2].
- Extend the algorithm to the k-colourable subgraph problem.

---
