Greedy algorithms for finding a set of generators

Question: *For a given vector space $\mathcal{V}$, what is the minimum number of vectors whose span equals $\mathcal{V}$?*

How can we obtain a minimum number of vectors?

Two natural approaches come to mind, the *Grow* algorithm and the *Shrink* algorithm.

Grow algorithm

def $\textsc{Grow}(\mathcal{V})$
  $S = \emptyset$
  repeat while possible:
    find a vector $\mathbf{v}$ in $\mathcal{V}$ that is not in Span S, and put it in S.

The algorithm stops when there is no vector to add, at which time S spans all of $\mathcal{V}$.
Thus, if the algorithm stops, it will have found a generating set.

But is it bigger than necessary?

Shrink Algorithm

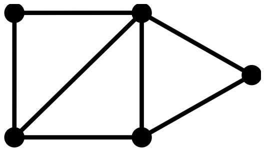
def $\textsc{Shrink}(\mathcal{V})$
  S = some finite set of vectors that spans $\mathcal{V}$
  repeat while possible:
    find a vector **v** in S such that Span $(S - \{v\}) = \mathcal{V}$, and remove **v** from S.

The algorithm stops when there is no vector whose removal would leave a spanning set.
At every point during the algorithm, S spans $\mathcal{V}$, so it spans $\mathcal{V}$ at the end.
Thus, if the algorithm stops, the algorithm will have found a generating set.

The question is, again: is it bigger than necessary?

When greed fails

Is it obvious that Grow algorithm and Shrink algorithm find smallest sets of generators? Look at example for a problem in *graphs*...

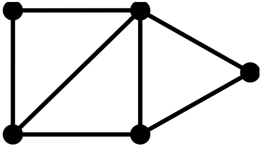


Points are called *nodes*, links are called *edges*.
Each edge has two *endpoints*, the nodes it connects. The endpoints of an edge are *neighbors*.

Definition: A *dominating set* in a graph is a set S of nodes such that every node is in S or a neighbor of a node in S.

When greed fails: dominating set

Definition: A *dominating set* in a graph is a set S of nodes such that every node is in S or a neighbor of a node in S.



Grow Algorithm: initialize $S = \emptyset$ while S is not a dominating set,   add a node to S.

Shrink Algorithm: initialize $S =$ all nodes while there is a node x such that $S - \{x\}$ is a dominating set,   remove x from S

Neither algorithm is guaranteed to find the smallest solution.