Word Ladder APT

- From->[words]->to
  - From hit to cog via [hot,dot,lot,dog,log]
- What words reachable from 'from'?
  - Repeat until we get to 'cog'
- Problem: reachable from 'dot'
  - Why not include 'hot'?
  - Don't re-use words
- Algorithm:
  - Find all words 1-away
  - From each n-away find (n+1)-away

Digression: word ladders

- How many ladders from cart to dire as shown?
  - Enqueue dare more than once?
  - Downside? Alternative?
- We want to know number of ladders that end at W.
  - What do we know initially?
  - When we put something on the queue, what do we know?
  - How do we keep track?
- Initialize and update per-word statistics

Word Ladder: more details

- # ladders that end at dare
  - At each word W
- Ladder length to W
  - Calculable from??
- Two maps

Dequeue s
  - foreach W one-away
  - if not-seen ???
  - else ???

Alan Kay

- Turing award 2003
  - OO programming, Dynabook
- “The best way to predict the future is to invent it”
- “American’s have no past and no future, they live in an extended present.”

I think the main thing about doing ...any kind of programming work, is that there has to be some exquisite blend between beauty and practicality. There’s no reason to sacrifice either one of those, and people who are willing to sacrifice either one of those, I don’t think really get what computing is all about.