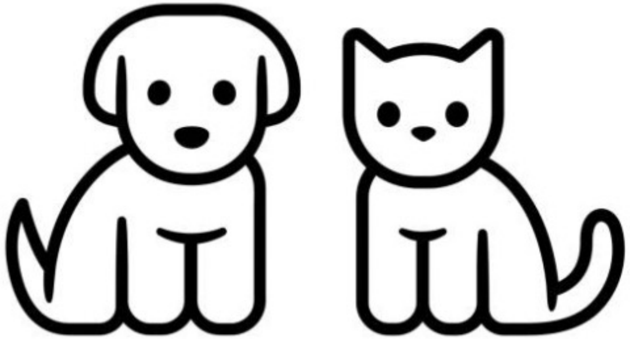
HW2 heuristic help

FED => BED ... => BOX

- Consider finding optimal paths from **FED** to **BOX**
- Assume the first step is **FED => BED**
- How can our **heuristic** estimate the cost to go from **BED to BOX**?
- If self.cost == ’steps’:
  - At least two letters much change
- If self.cost == ’scrabble’:
  - The 2nd letter eventually mut be changed to an O
  - The 3rd letter eventually mut be changed to an X
- If self.cost == ’frequency’:
  - At least two letters must change (each with cost 1 + something more)
  - We’ll need to use the word BOX (incurring a cost of ?)
  - We’ll need to use some word with an O in position 1 (with minimum cost of ?)



```
>> python dcsolve.py fed box steps
    dc(fed,box,steps) cost:3.00; time:0.000;
    solution:fed bed bod box; deltas:[0.0, 0.0, 0.0]; ADMISSIBLE
>> python dcsolve.py fed box scrabble
    • dc(fed,box,scrabble) cost:12.00; time:0.000;
      solution:fed bed bod box; deltas:[0.0, 0.0, 0.0]; ADMISSIBLE
>> python dcsolve.py fed box frequency
    dc(fed,box,frequency) cost:23.15; time:0.068;
    solution:fed few new now bow box; deltas:[-15.21, -11.609, -
    9.274, -7.698, 0.0]; ADMISSIBLE
```