

admissible
heuristics
26 sept
2022

CSCI
373

if H is consistent,
then A^* is
optimal

if H is admissible,
then non-memoized A^* is
optimal

but how do we design such heuristics?

back to sliding tiles



what might be an
admissible heuristic?

7	2	4
5		6
8	3	1

$$H(q)$$

heuristic
function



$$H^*(q_0) \leq \text{optimal completion cost}$$

1	2	3
4	5	6
7	8	

final
state

your answer here

7	2	4
5	6	
8	3	1



$$H^*(q_0)$$

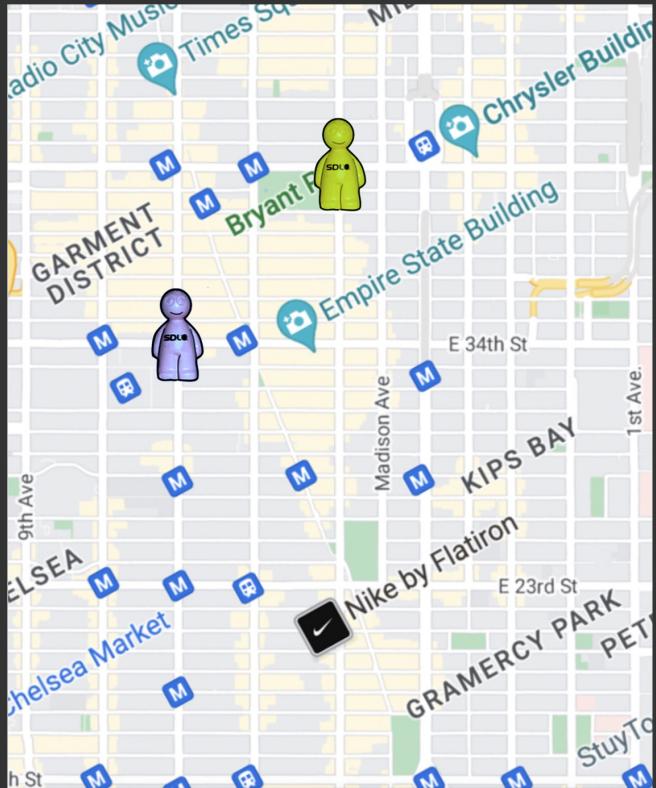
1	2	3
4	5	6
7	8	

final
state

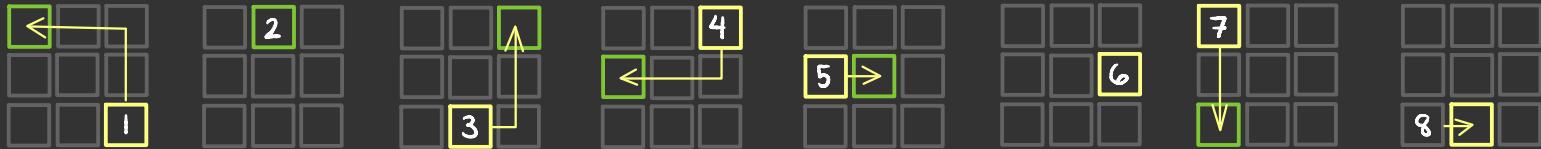
$$H(q) = 6$$

<

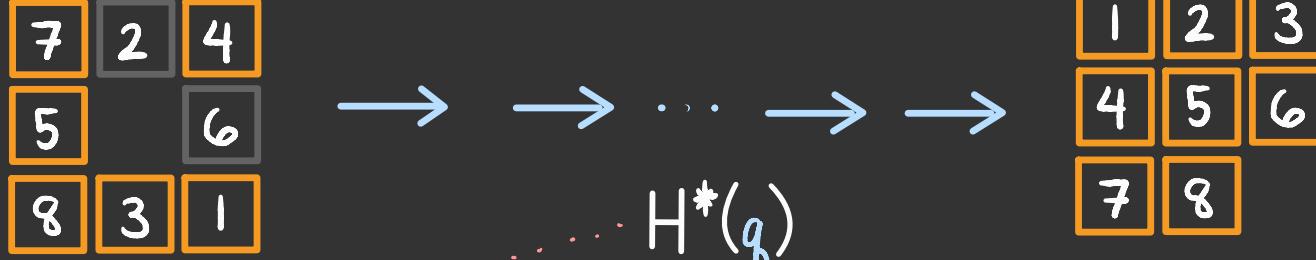
number of misplaced tiles



manhattan distance



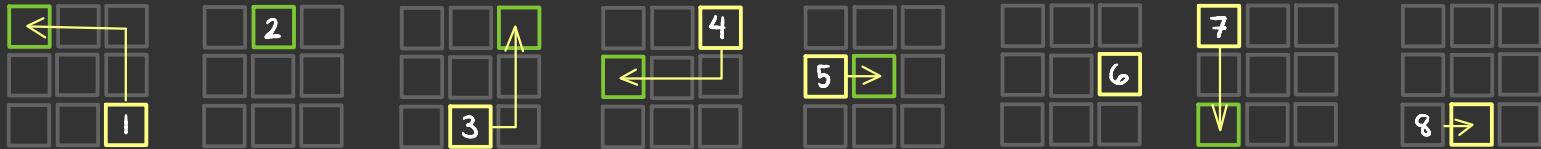
4 + 0 + 3 + 3 + | + 0 + 2 + |



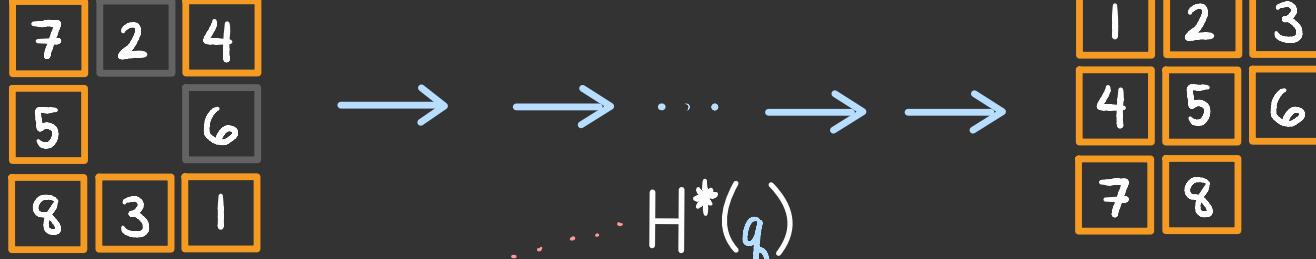
$H(q) = 14 \leq \dots$

cumulative
manhattan distance between current
locations and target locations

final
state



4 + 0 + 3 + 3 + | + 0 + 2 + |



$$H(q) = 14$$

cumulative

final
state

why is
this ?
admissible?

manhattan distance between current
locations and target locations



$$4 + 0 + 3 + 3 + | + 0 + 2 + | + 2$$



$H(q) = 14 \leq \dots$

final
state

what if we include the empty space?

1	2	3
4	5	6
7	8	



$$H^*(q_0) = 1$$

$$H(q) = 2$$

X

not
admissible

1	2	3
4	5	6
7	8	

final
state

what if we include the empty space?

is there a less ethereal way to derive

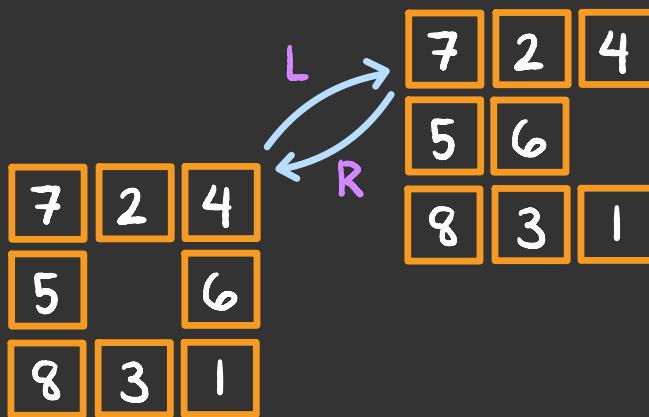
admissible heuristics?



relax

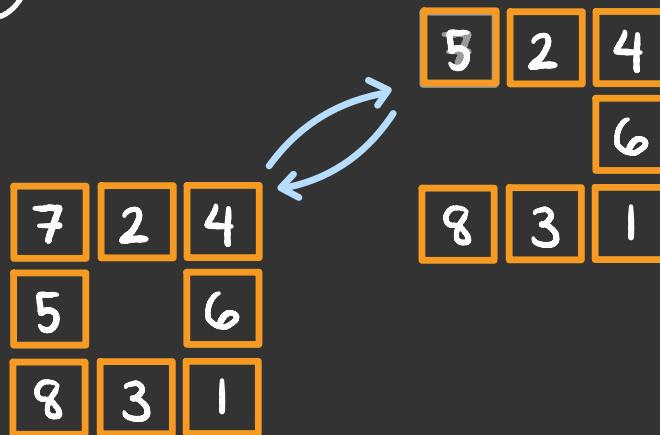
a tile can move from square A to square B if:

- A is adjacent to B
- B is empty



a tile can move from square A to square B if:

- A is adjacent to B
- ~~B is empty~~

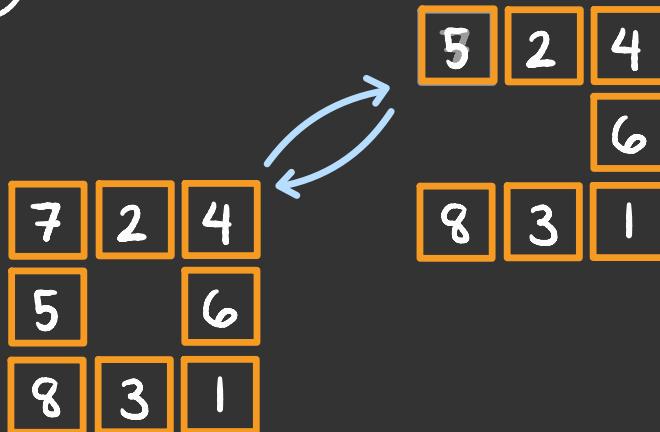


a tile can move from square A to square B if:

- A is adjacent to B

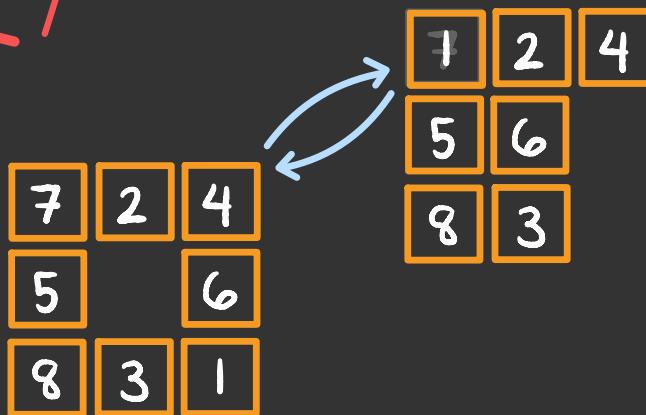
~~• B is empty~~

the optimal solution to this "relaxed" puzzle is cumulative manhattan distance



a tile can move from square A to square B if:

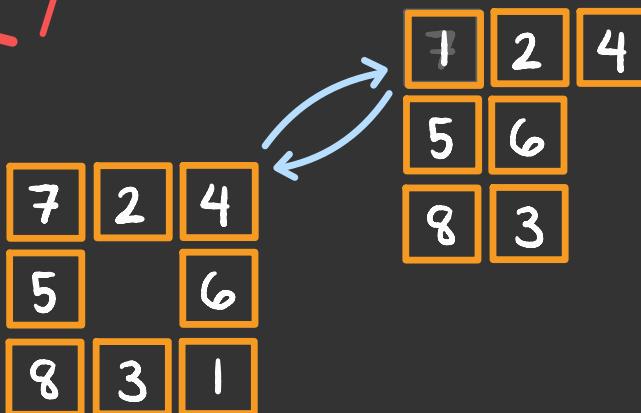
- A is adjacent to B
- B is empty



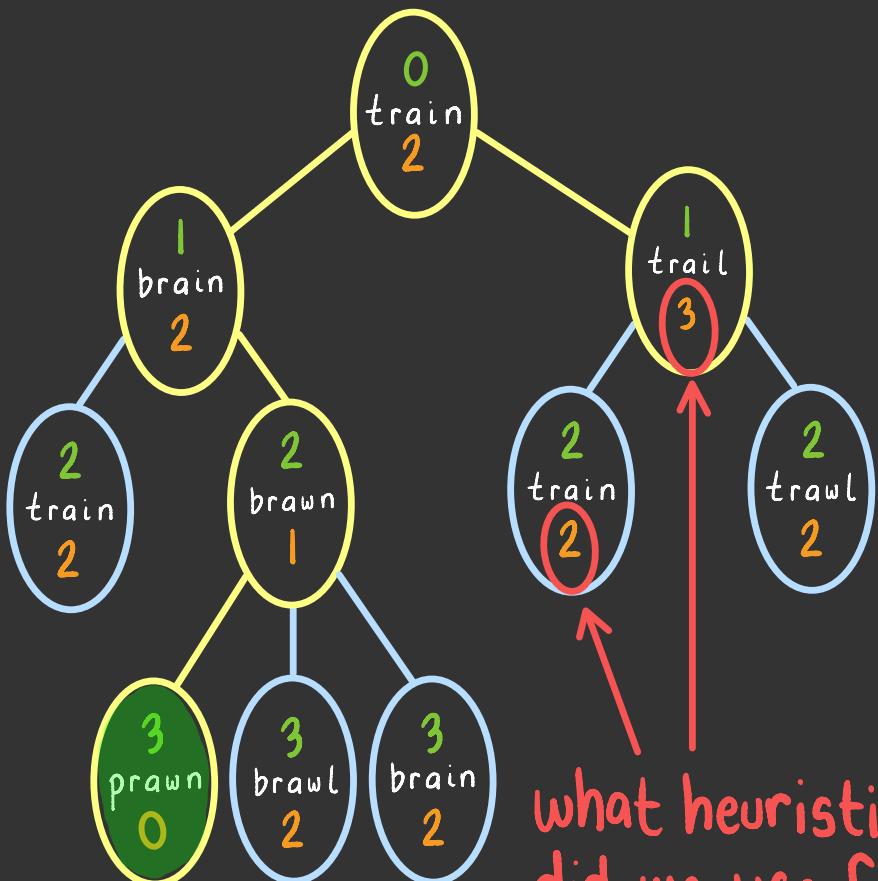
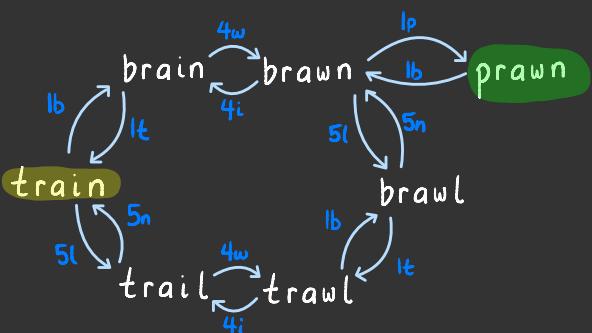
a tile can move from square A to square B if:

- A is adjacent to B
- B is empty

the optimal solution to this "relaxed" puzzle is number of misplaced tiles

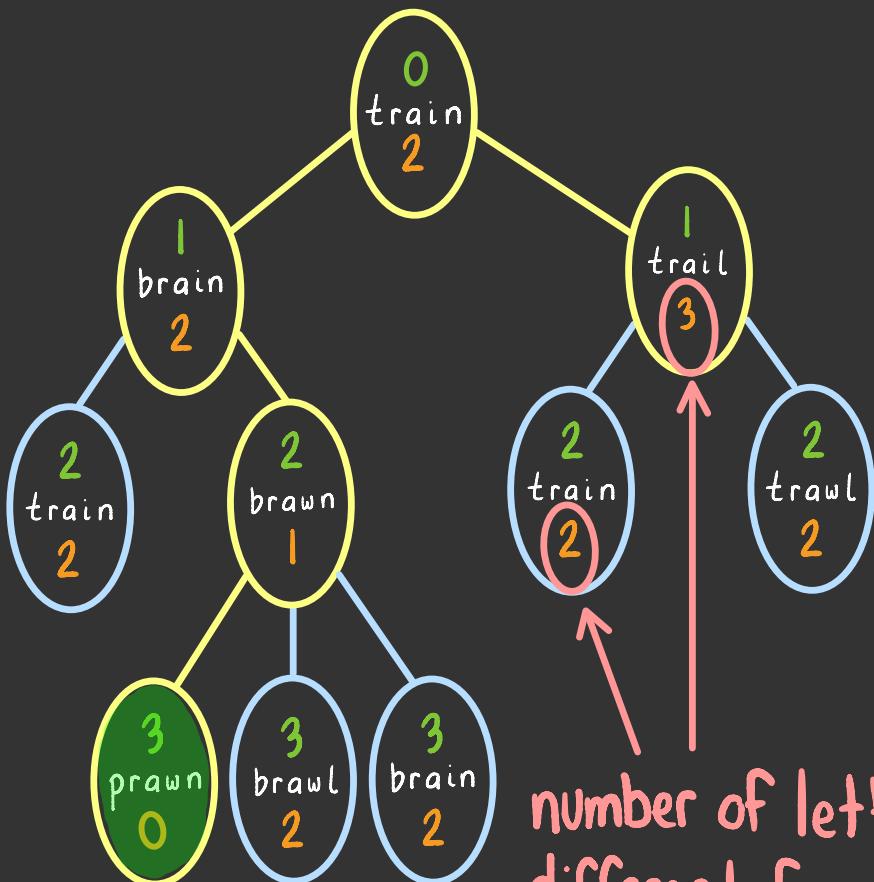
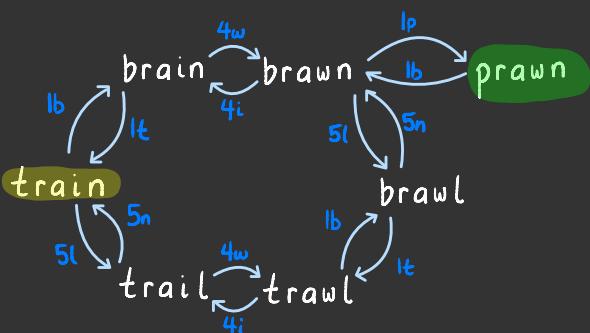


State machine:



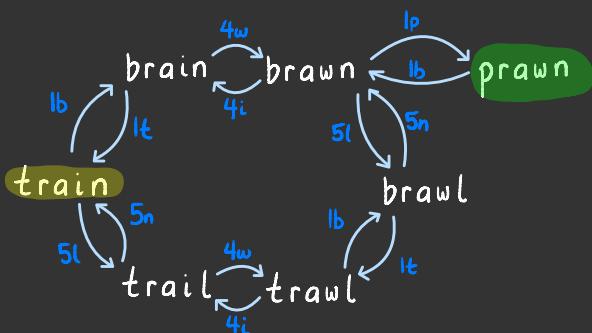
what heuristic
did we use for
word ladder?

State machine:



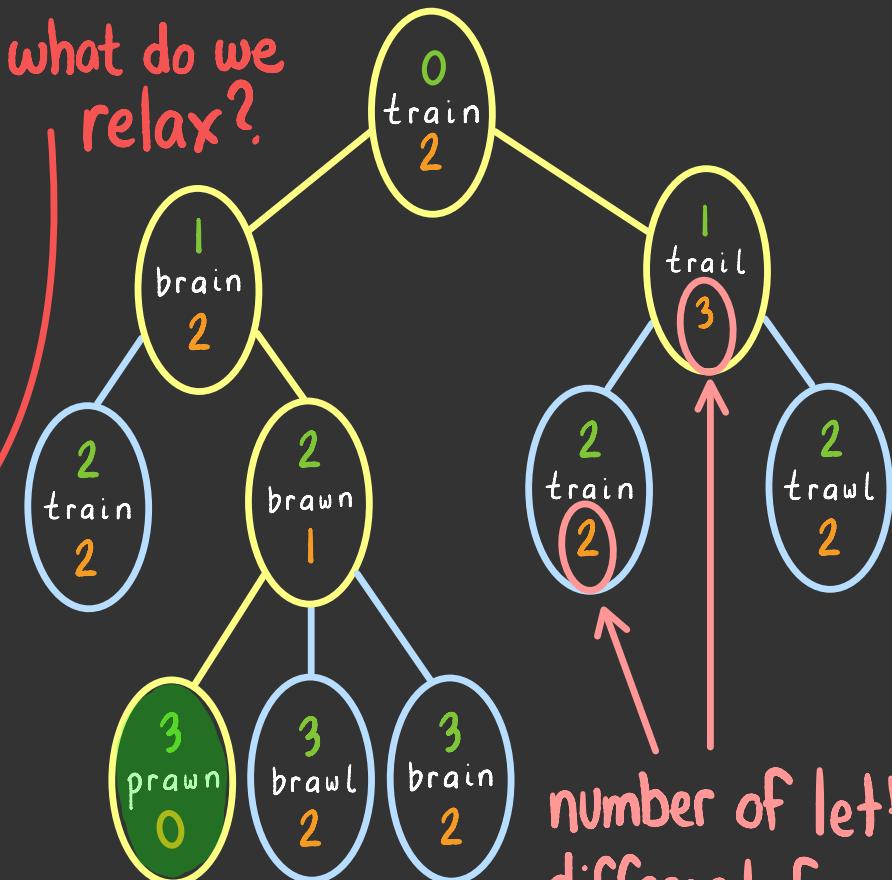
number of letters
different from the
target word

State machine:



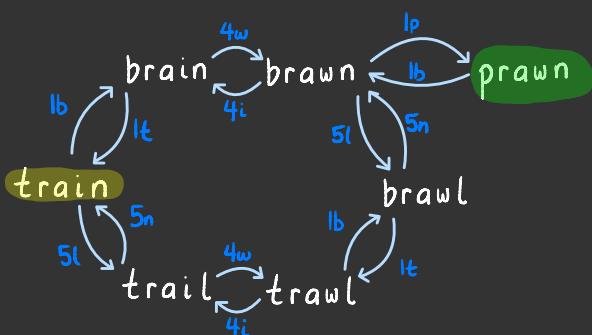
we can move from one word to any other english word that is one letter different

what do we relax?

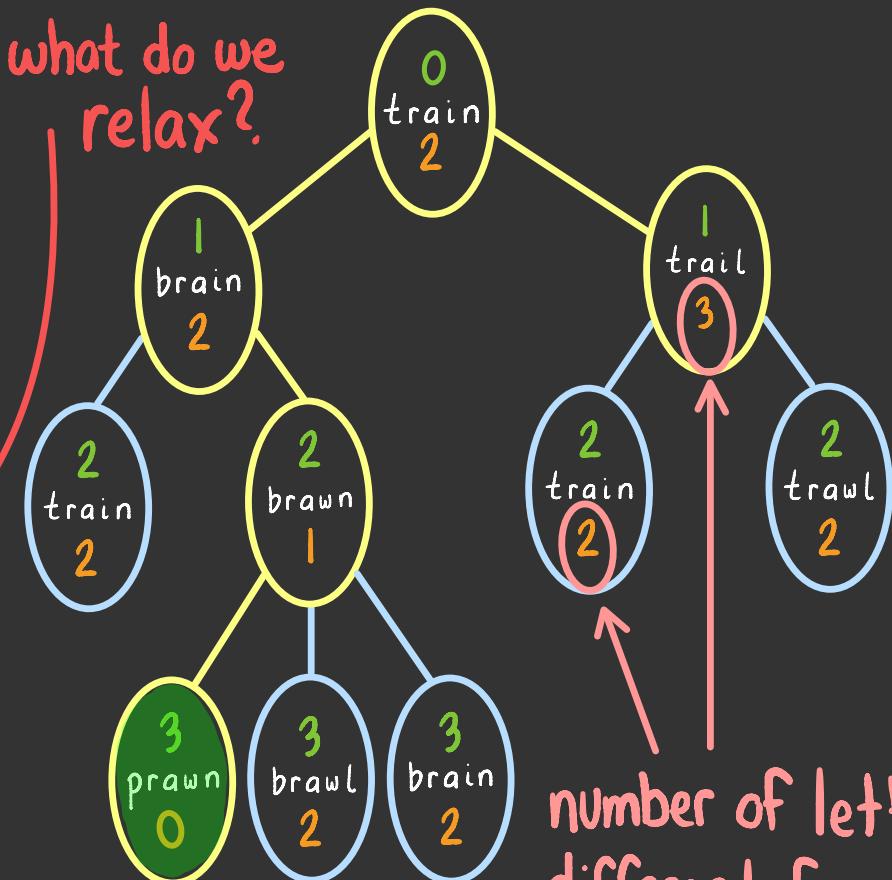


number of letters different from the target word

State machine:



what do we relax?



we can move from one word to any ~~other~~ string ~~english word~~ that is one letter different

number of letters different from the target word