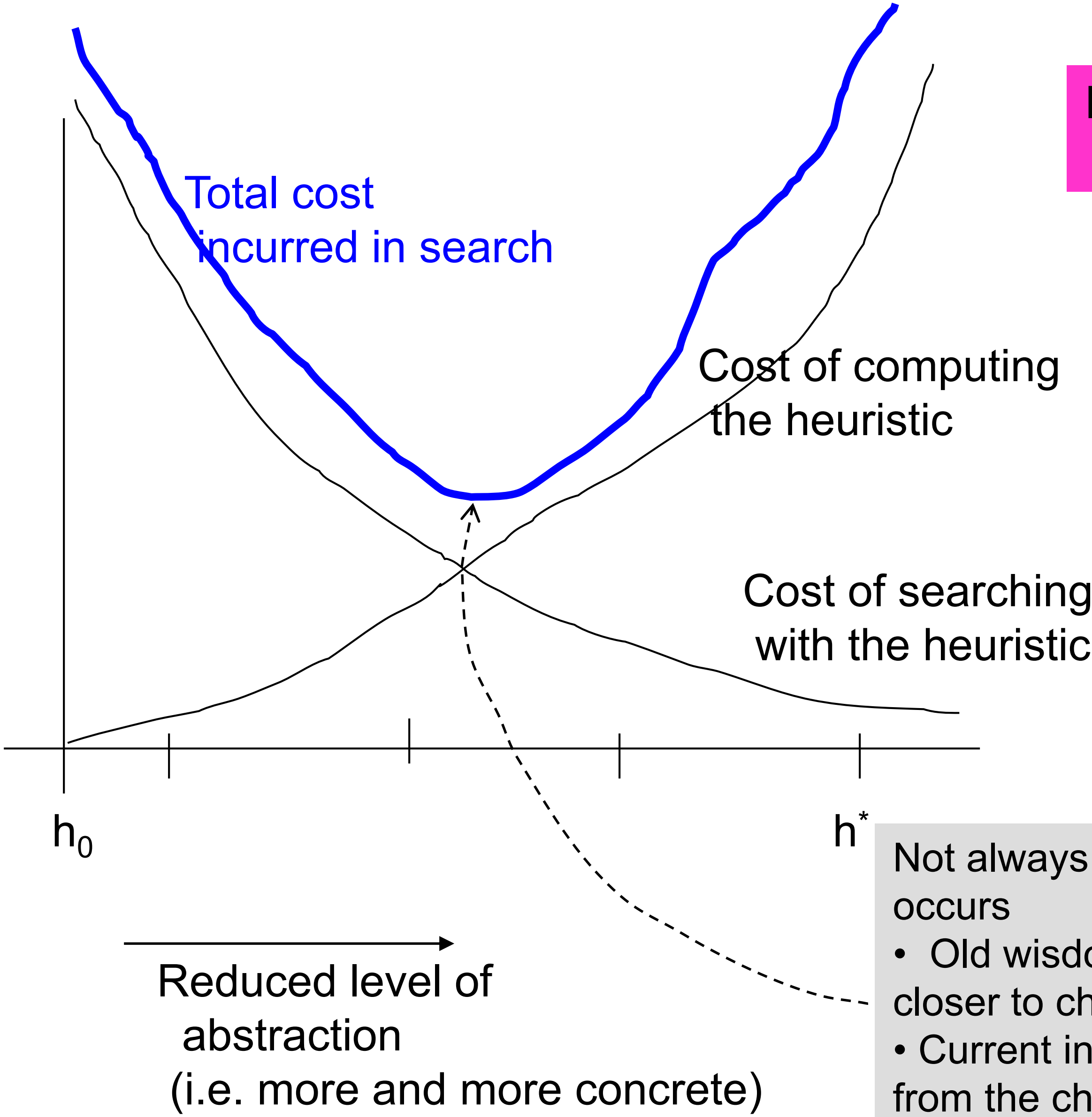
How informed should the heuristic be?




Not always clear where the total minimum occurs

- Old wisdom was that the global min was closer to cheaper heuristics
- Current insights are that it may well be far from the cheaper heuristics for many problems
  - E.g. Pattern databases for 8-puzzle
  - Plan graph heuristics for planning

# Sizes of Problem Spaces

| Problem | Nodes | Brute-Force Search Time (10 million nodes/second) |
| --- | --- | --- |
| • 8 Puzzle: | $10^5$ | .01 seconds |
| • $2^3$ Rubik's Cube: | $10^6$ | .2 seconds |
| • 15 Puzzle: | $10^{13}$ | 6 days |
| • $3^3$ Rubik's Cube: | $10^{19}$ | 68,000 years |
| • 24 Puzzle: | $10^{25}$ | 12 billion years |

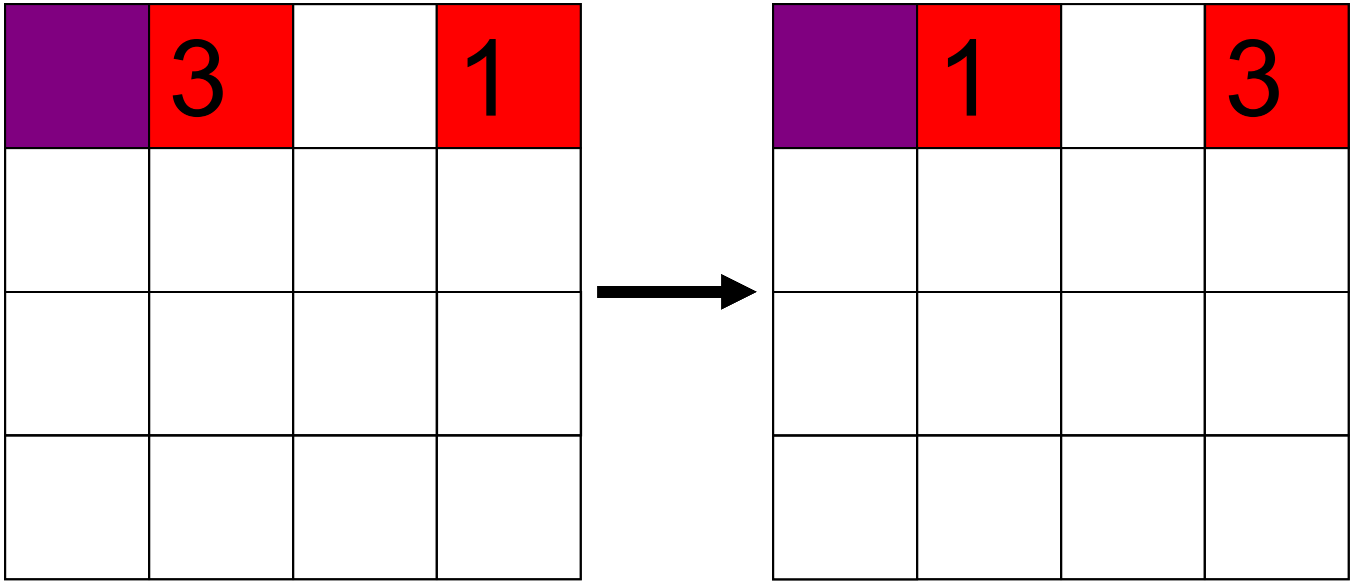
# Performance of IDA* on 15 Puzzle

- Random 15 puzzle instances were first solved optimally using IDA* with Manhattan distance heuristic (Korf, 1985).
- Optimal solution lengths average 53 moves.
- 400 million nodes generated on average.
- Average solution time is about 50 seconds on current machines.

# Limitation of Manhattan Distance

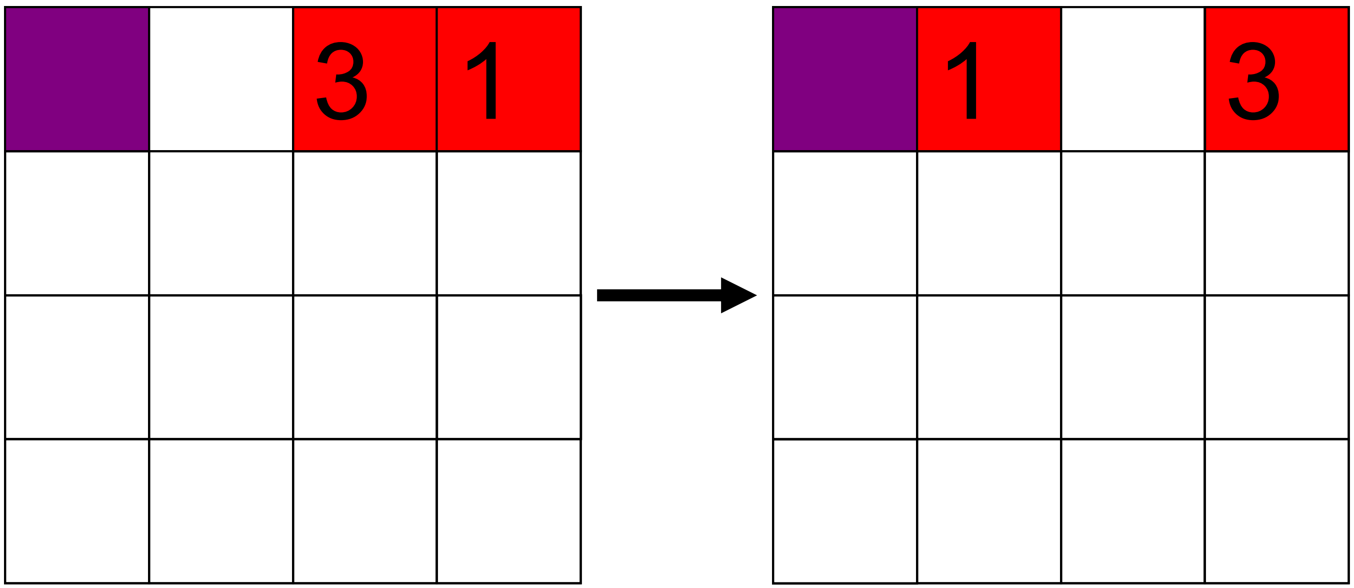
- To solve a 24-Puzzle instance, IDA* with Manhattan distance would take about 65,000 years on average.
- Assumes that each tile moves independently
- In fact, tiles interfere with each other.
- Accounting for these interactions is the key to more accurate heuristic functions.

# Example: Linear Conflict



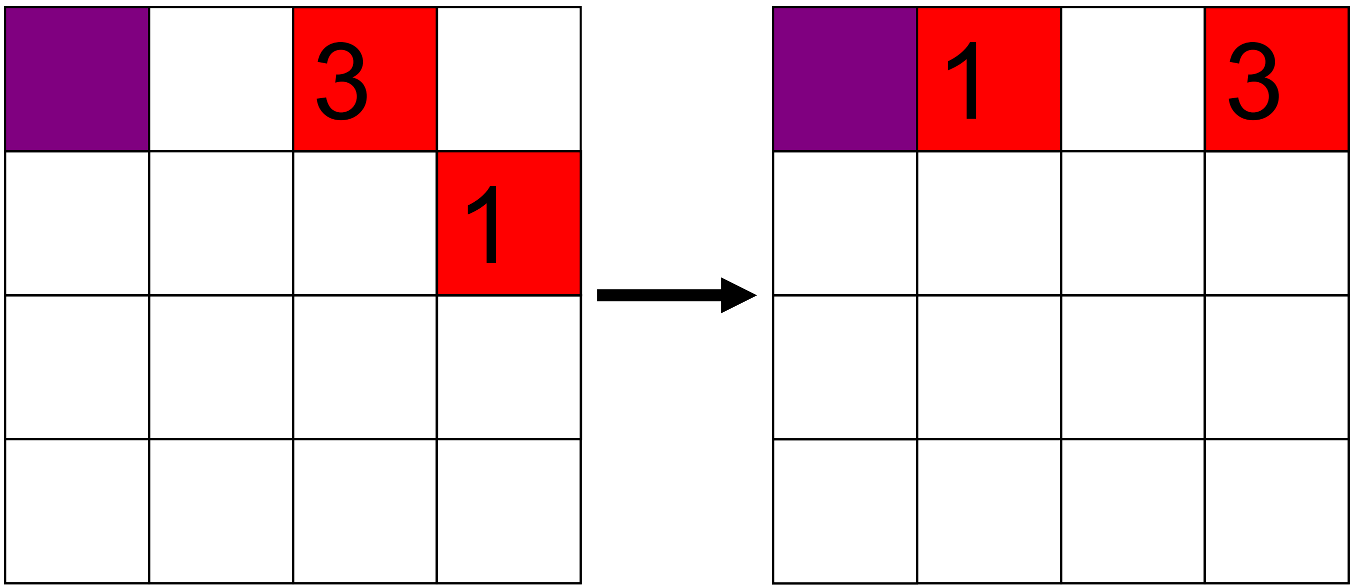

Manhattan distance is 2+2=4 moves

# Example: Linear Conflict



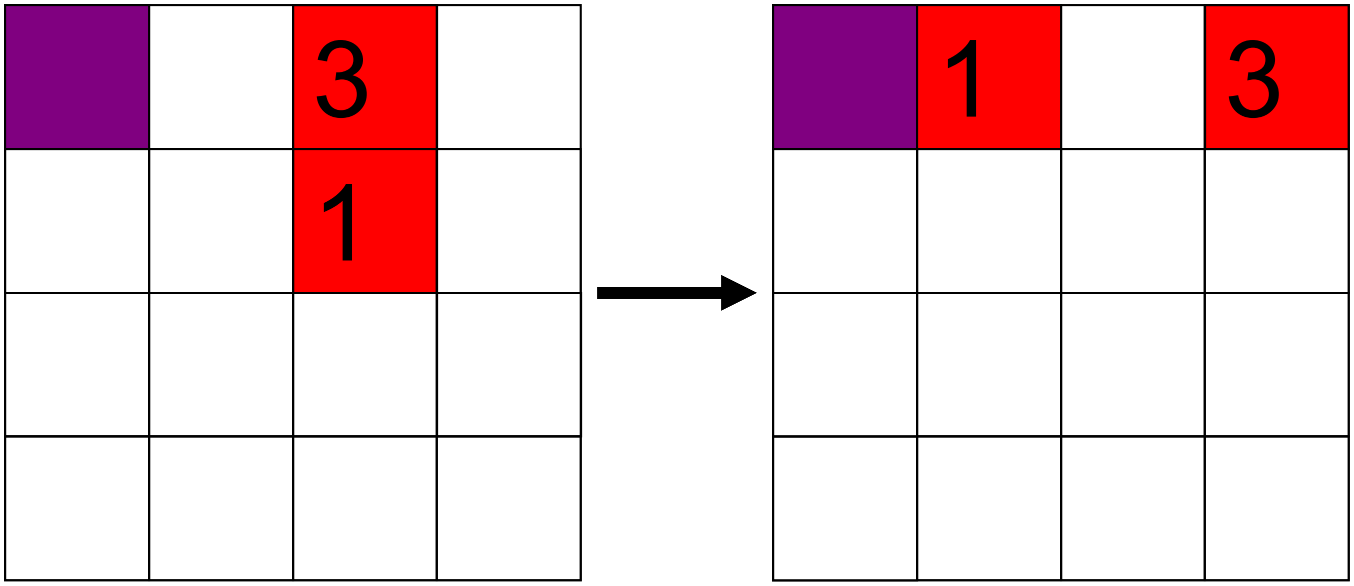

Manhattan distance is 2+2=4 moves

# Example: Linear Conflict



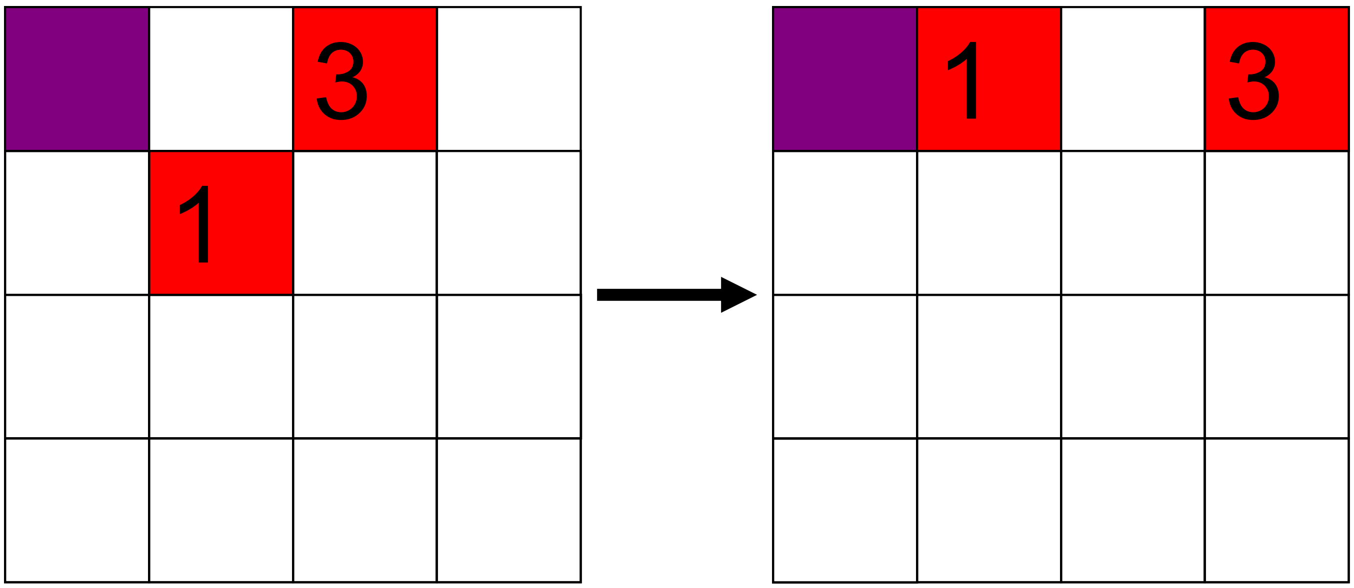

Manhattan distance is 2+2=4 moves

# Example: Linear Conflict



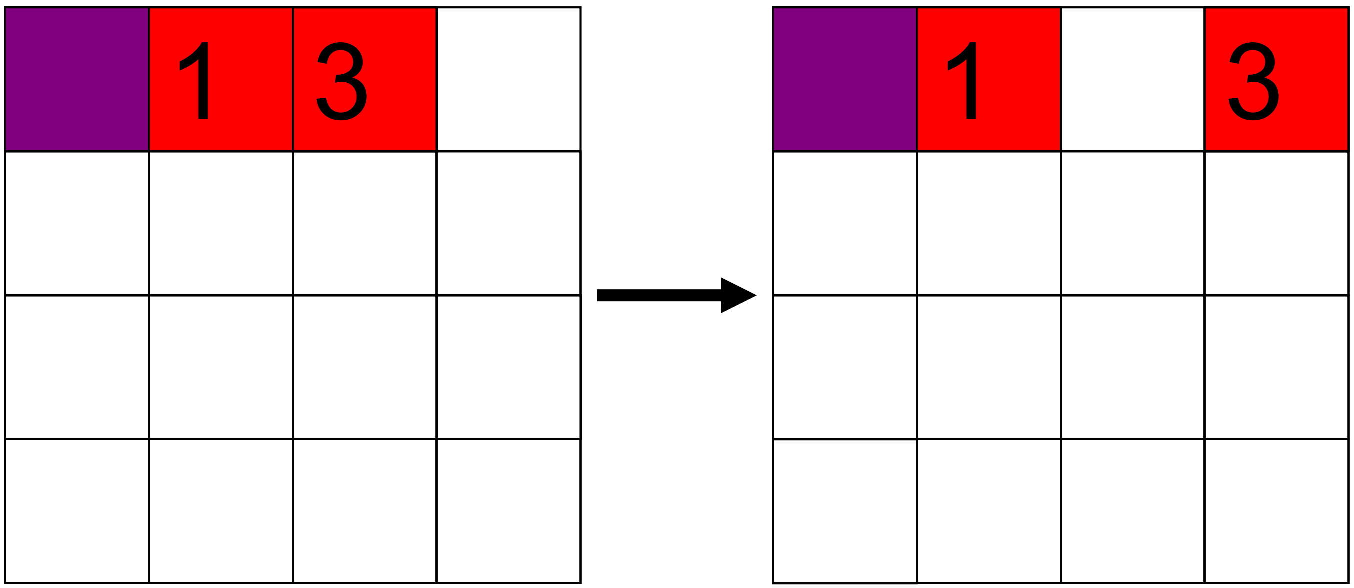

Manhattan distance is 2+2=4 moves

# Example: Linear Conflict



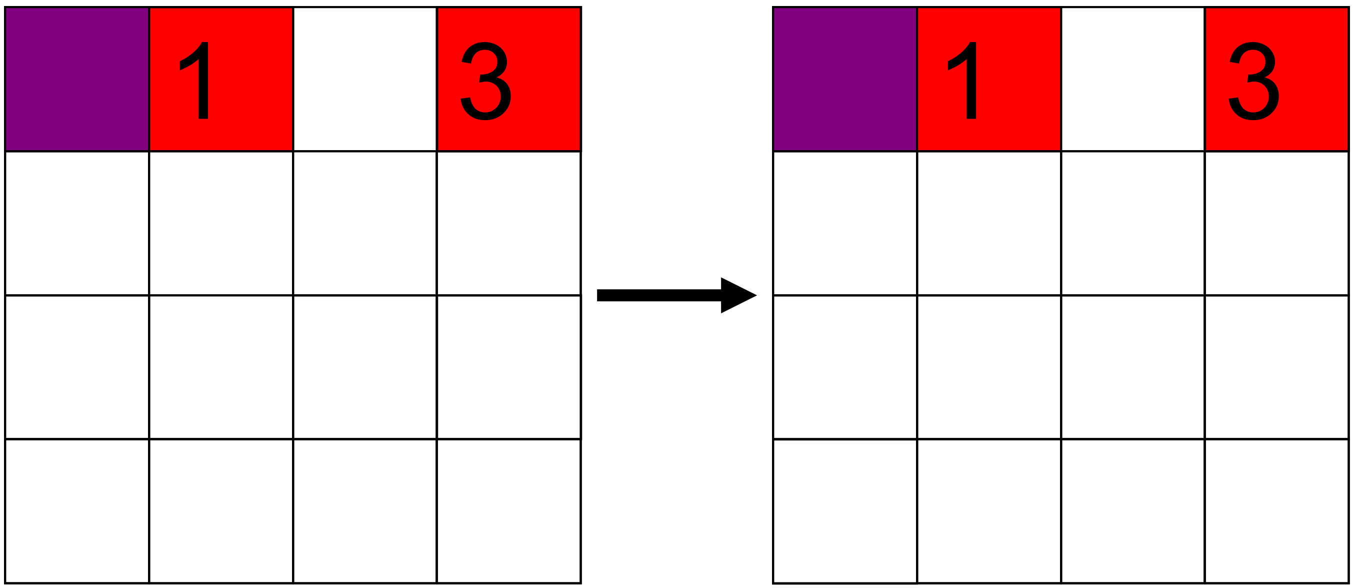

Manhattan distance is 2+2=4 moves

# Example: Linear Conflict




Manhattan distance is 2+2=4 moves

# Example: Linear Conflict

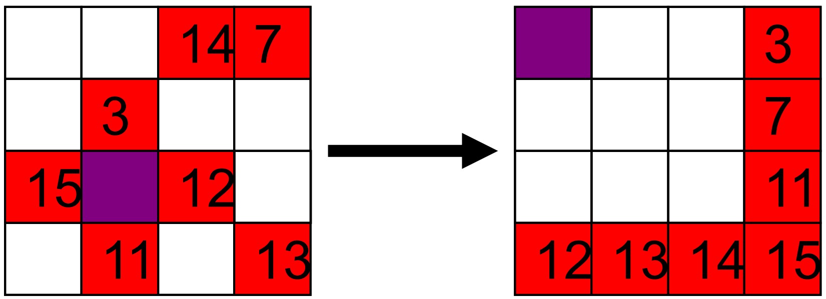



Manhattan distance is 2+2=4 moves, but linear conflict adds 2 additional moves.

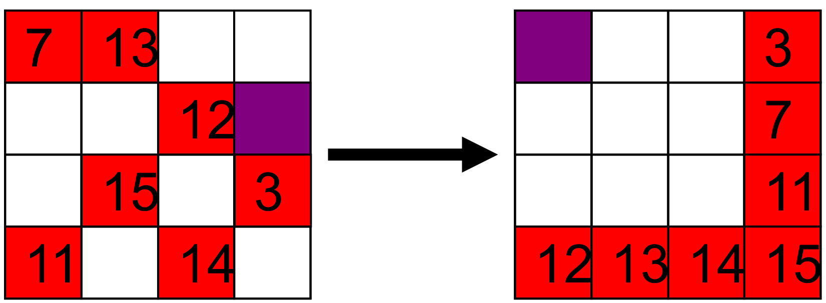
# Linear Conflict Heuristic

- Hansson, Mayer, and Yung, 1991
- Given two tiles in their goal row, but reversed in position, additional vertical moves can be added to Manhattan distance.
- Still not accurate enough to solve 24-Puzzle
- We can generalize this idea further.

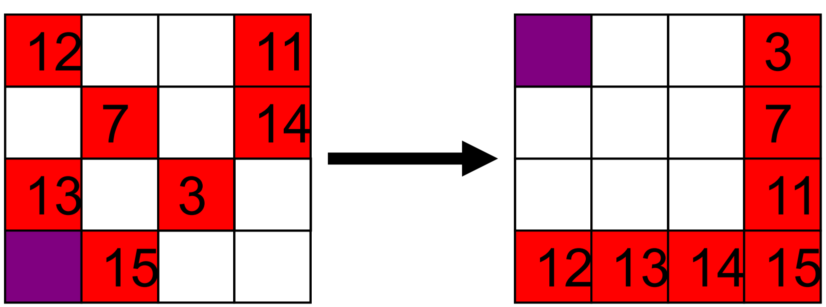
# More Complex Tile Interactions




M.d. is 19 moves, but 31 moves are needed.




M.d. is 20 moves, but 28 moves are needed

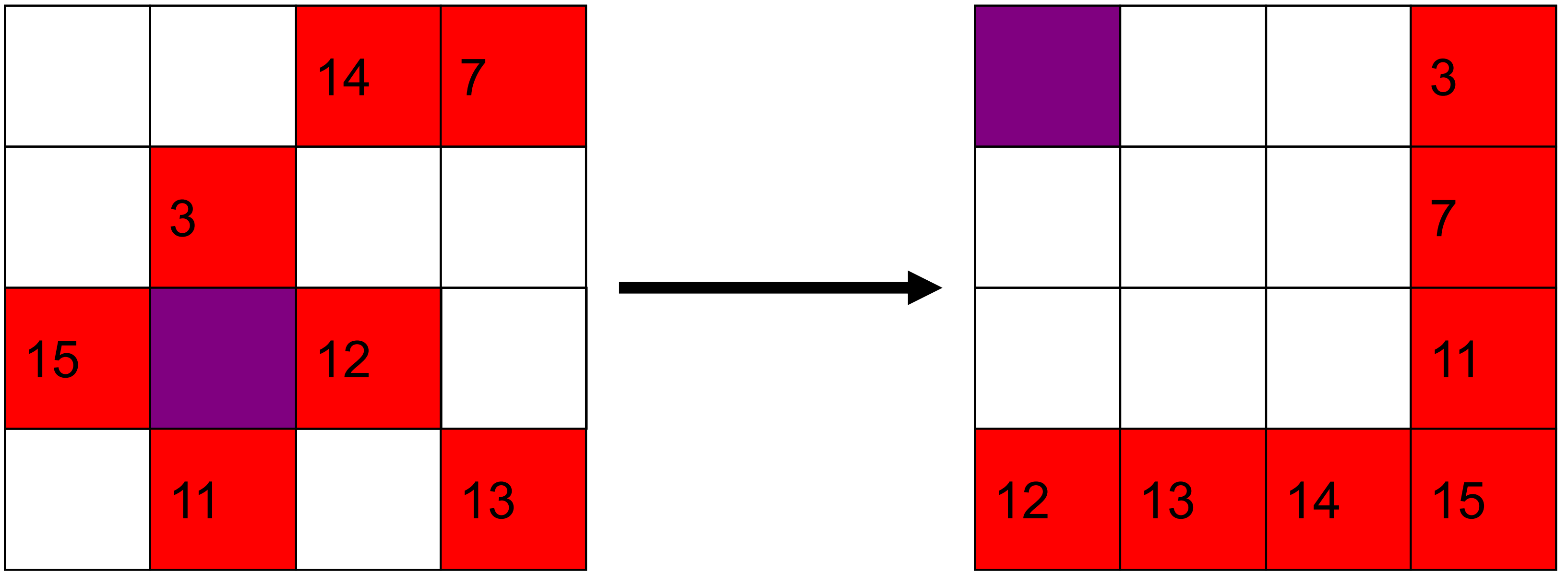



M.d. is 17 moves, but 27 moves are needed

# Pattern Database Heuristics

- Culberson and Schaeffer, 1996
- A pattern database is a complete set of such positions, with associated number of moves.
- e.g. a 7-tile pattern database for the Fifteen Puzzle contains 519 million entries.

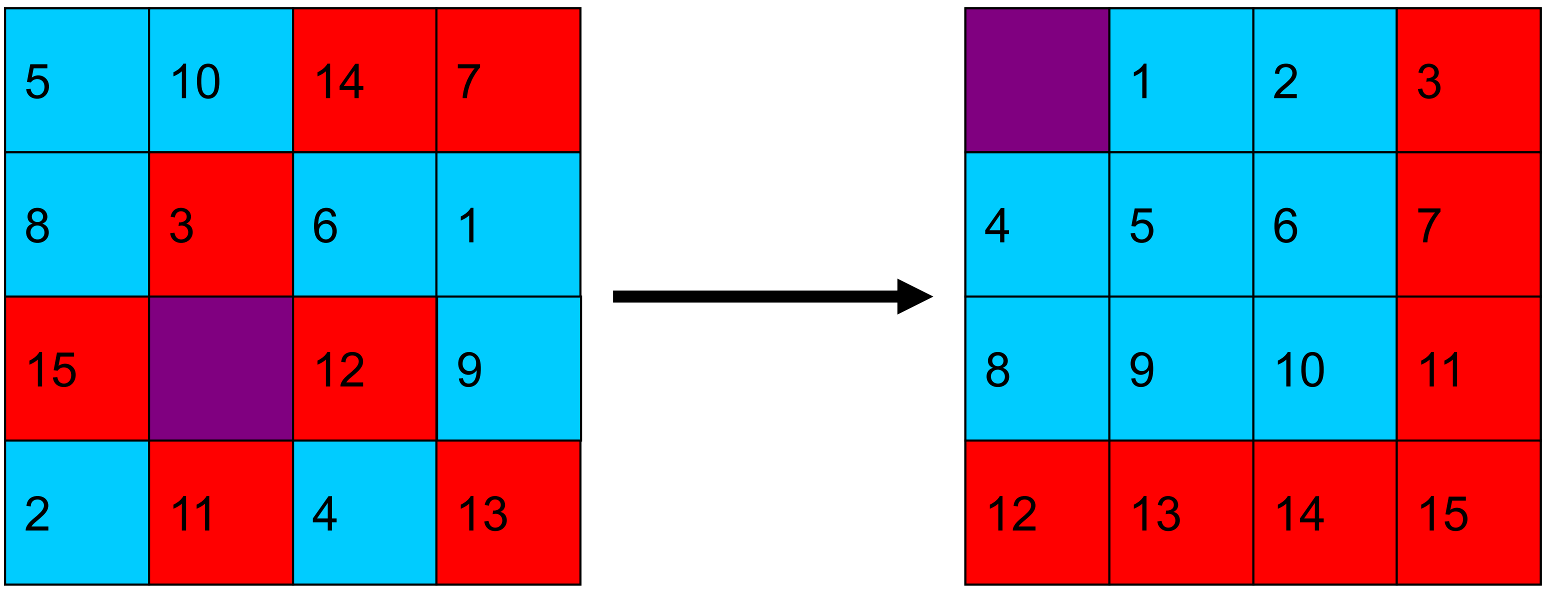
# Example 8-tile pattern



# Precomputing Pattern Databases

- Entire database is computed with one backward breadth-first search from goal.
- All non-pattern tiles are indistinguishable, but all tile moves are counted.
- The first time each state is encountered, the total number of moves made so far is stored.
- Once computed, the same table is used for all problems with the same goal state.

# Combining Multiple Databases




31 moves needed to solve red tiles

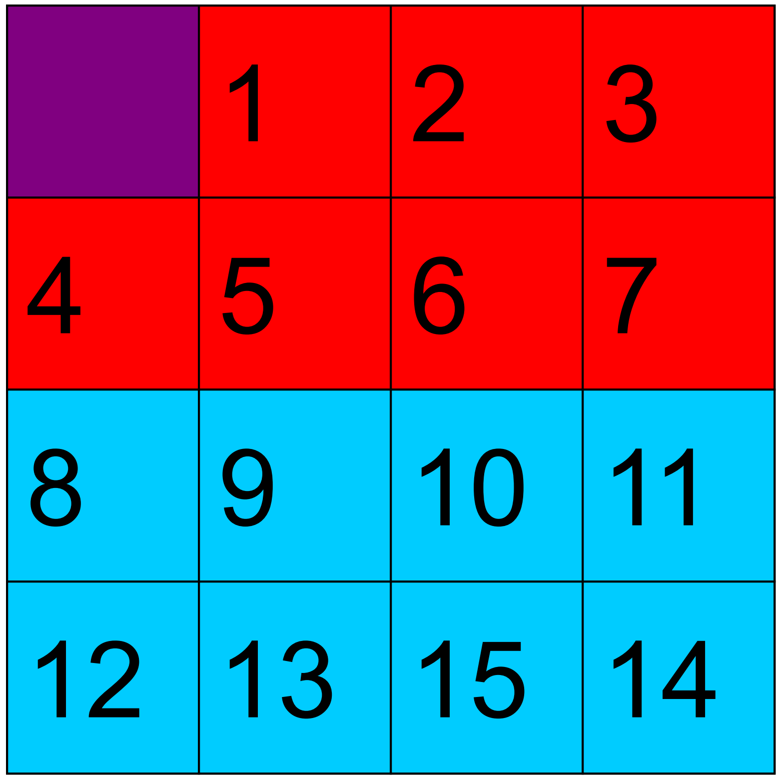
22 moves need to solve blue tiles

Overall heuristic is maximum of 31 moves

# Additive Pattern Databases

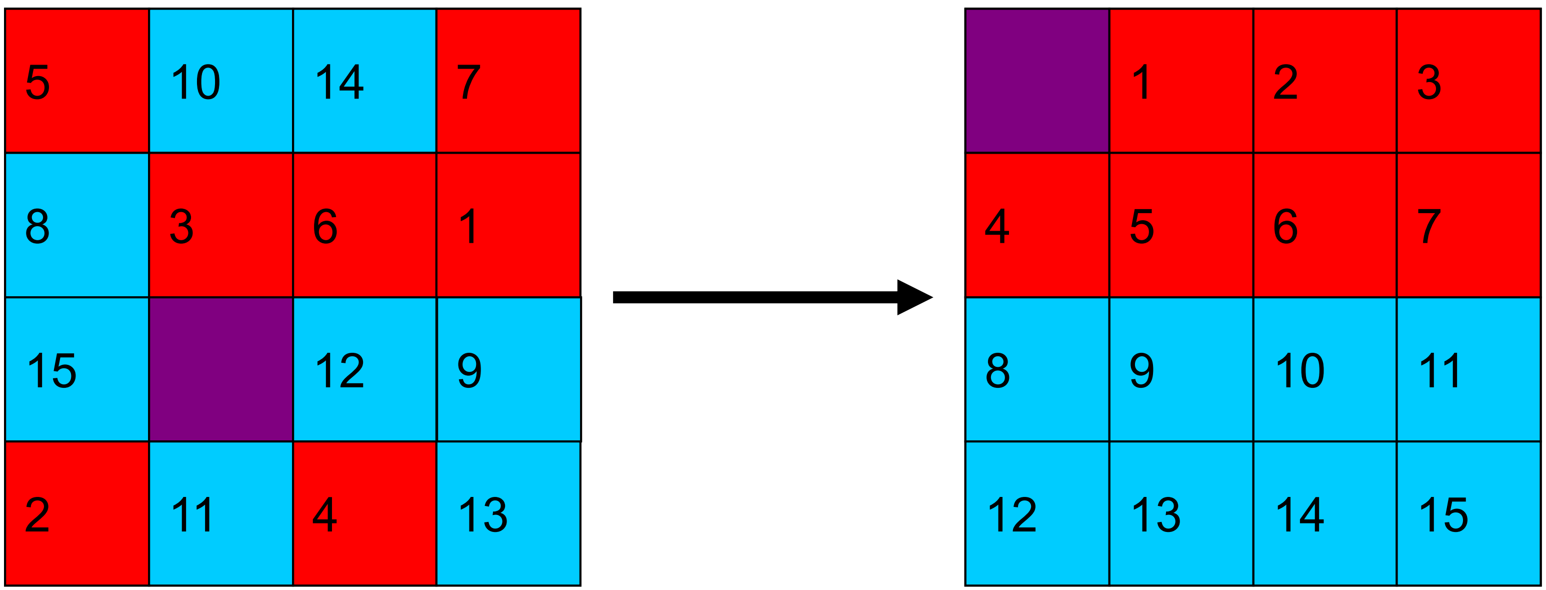
- Culberson and Schaeffer counted all moves needed to correctly position the pattern tiles.
- In contrast, we count only moves of the pattern tiles, ignoring non-pattern moves.
- If no tile belongs to more than one pattern, then we can add their heuristic values.
- Manhattan distance is a special case of this, where each pattern contains a single tile.

# Example Additive Databases




The 7-tile database contains 58 million entries. The 8-tile database contains 519 million entries.

# Computing the Heuristic




20 moves needed to solve red tiles

25 moves needed to solve blue tiles

Overall heuristic is sum, or 20+25=45 moves

# Performance

- 15 Puzzle: 2000x speedup *vs* Manhattan dist
  - IDA* with the two DBs shown previously solves 15 Puzzles optimally in 30 milliseconds

- 24 Puzzle: 12 million x speedup *vs* Manhattan
  - IDA* can solve random instances in 2 days.
  - Requires 4 DBs as shown
    - Each DB has 128 million entries
  - Without PDBs: 65,000 years