Day 7: Best-First Search

- Covering a few alternate search methods that are useful in AI
- Hill Climbing:
  - Simplest form of search, no storage needed for cycle checking
  - Highly parallellizable
  - Algorithm: Evaluate all possible moves, move in the "best" direction
    - Requires defining best
  - Alternative algorithm:
    - Repeat (until solution found or forever)
      - Add some random noise to the current solution
      - Keep the new solution IFF is better
  - Geographic metaphor:
    - A fitness function can impose a "landscape" over a problem/solution space
    - Then, we're looking for the peak, given that we can't see the whole landscape
  - Shape of the landscape matters, because if there are smaller peaks it can get stuck at a local maximum
    - Solutions can be random big jumps, heuristics, restarting, etc.
    - OTOH, if there are lots of solutions and finding the *best* solution doesn't matter, hill climbing can be great

      This image is on the desktop once Notion is connected to the internet
  - Originally shunned by AI but newer versions are making a comeback

- Heuristic search
  - Needs an easy way to approximately evaluate the "goodness" of a state, relative to the goal
  - Can't be precise, because then it would just be solving the problem
  - "Manhattan Distance"/"City Block Distance": A better evaluation distance
    - Informative, as smooth as possible
    - Ex (tile puzzle): Take the average distance each tile is from its destination
    - Easy to use: just add a manhattan sort to BFS (theoretically DFS could work too, but it's probably not worth it)
- Constraint Satisfaction
  - Common in perception
  - "Form of "expert" rapid problem solving"
  - uses parallelism
  - Hard to implement, but good when done well
  - Make inferences that are constraints, then try playing it out, and backtrack if needed
  - Successful use of constraint satisfaction: Waltz Labelling
    - Input is an ambigous 2D represnation of a 3D world, output is a 3D representation
    - Basically only works on vertices of blocks
    - Starts by labeling boundries, then uses constraints like the list of legal verticies, etc.