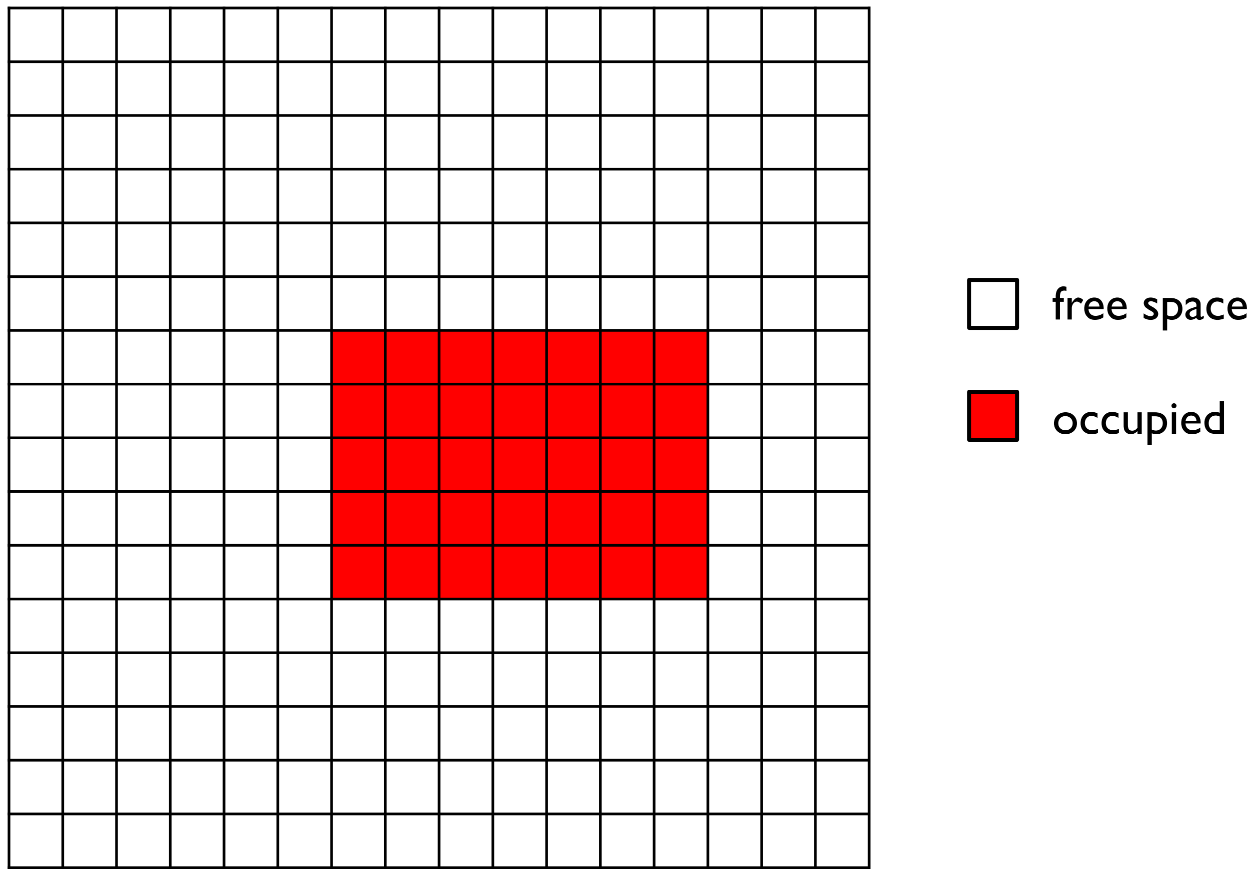
Path Planning in Discrete Sampled Space

Spatial Decomposition

- represent space itself, rather than the objects in it, using discrete samples
- many ways to perform sampling, but the simplest is to use a grid



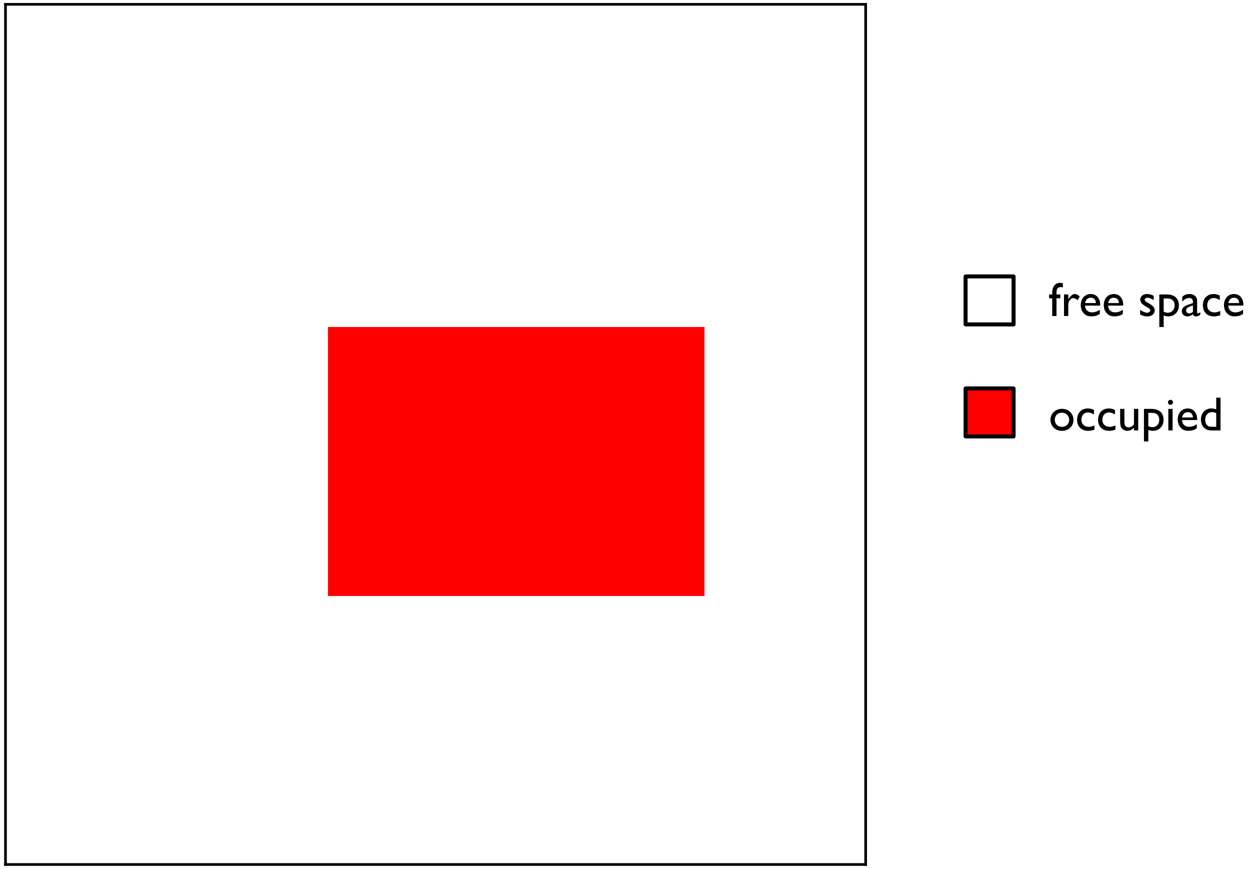
Uniform Spatial Sampling

- very general representation
  - grid locations can represent anything
- if something moves then the representation does not change dramatically
- limited by grid resolution
  - large cell size gives a coarse representation
  - small cell size is storage intensive
    - football pitch at 1cm^2 resolution
      - 105m x 68m x 100 x 100 = 71,400, 000 cells
    - 3D is much worse

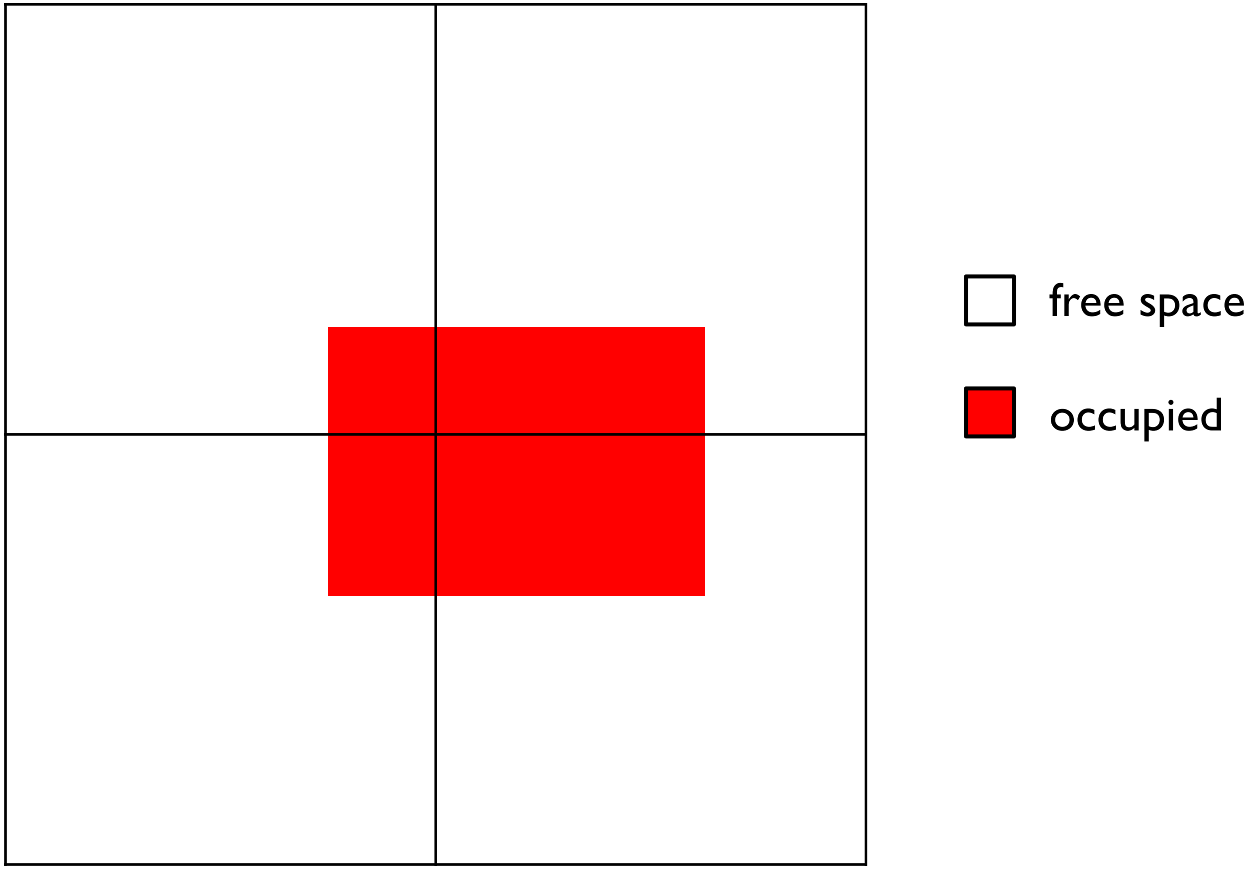
Recursive Hierarchical Representations

- storage space can be conserved by observing that free space cells and occupied cells tend to cluster
  - group the clusters into larger cells
- quadtree
  - recursively subdivide space into 4 equal-sized cells until every cell is either uniformly free or uniformly occupied
    - or some threshold resolution is reached

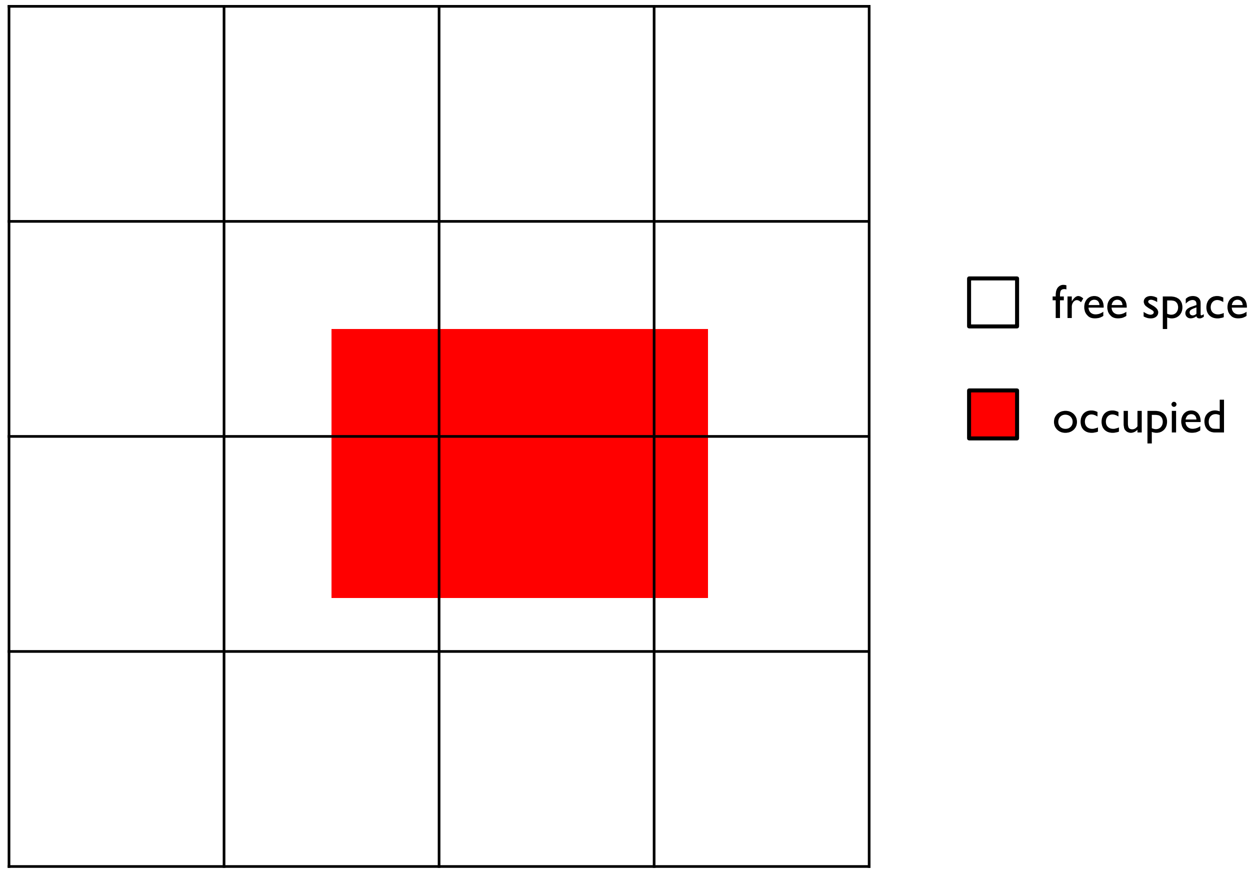
Quadtree Decomposition



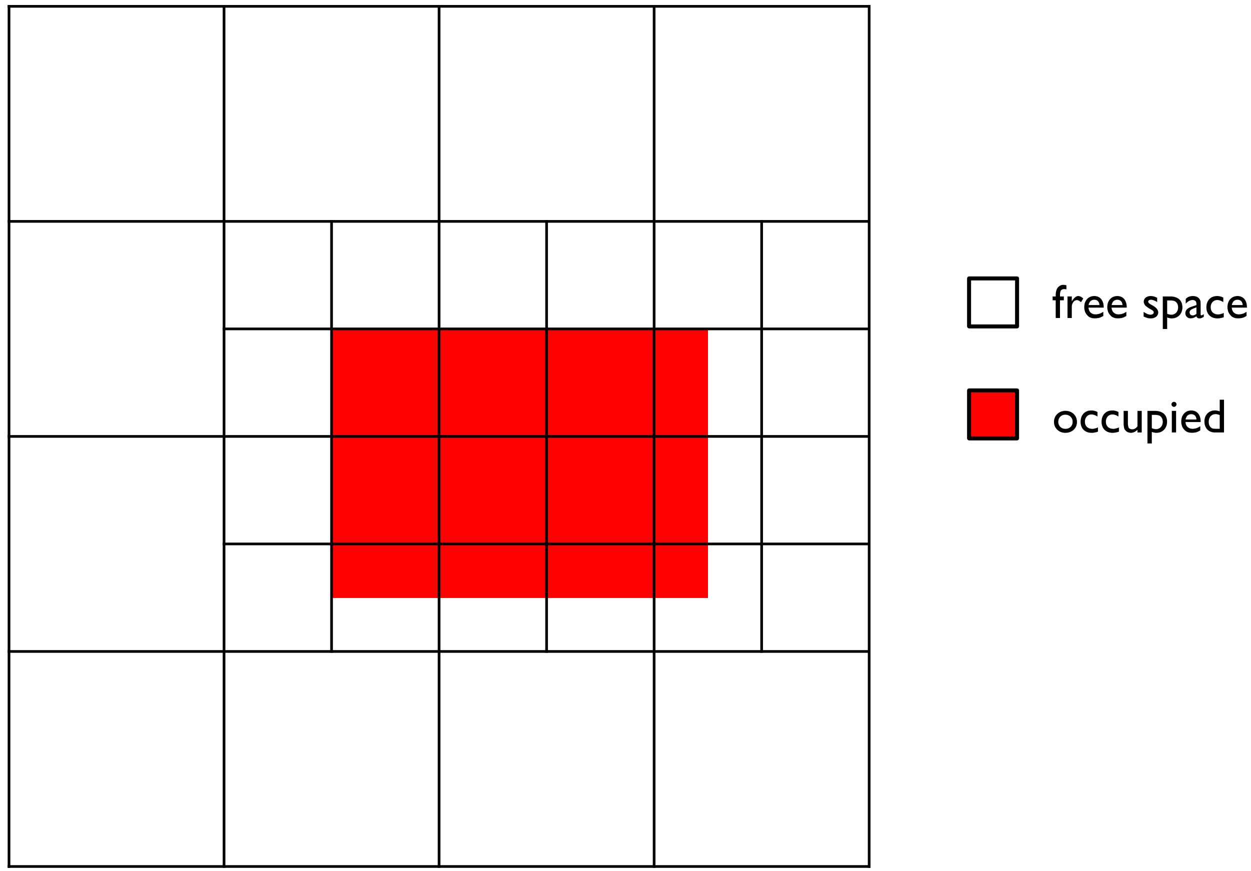
Quadtree Decomposition



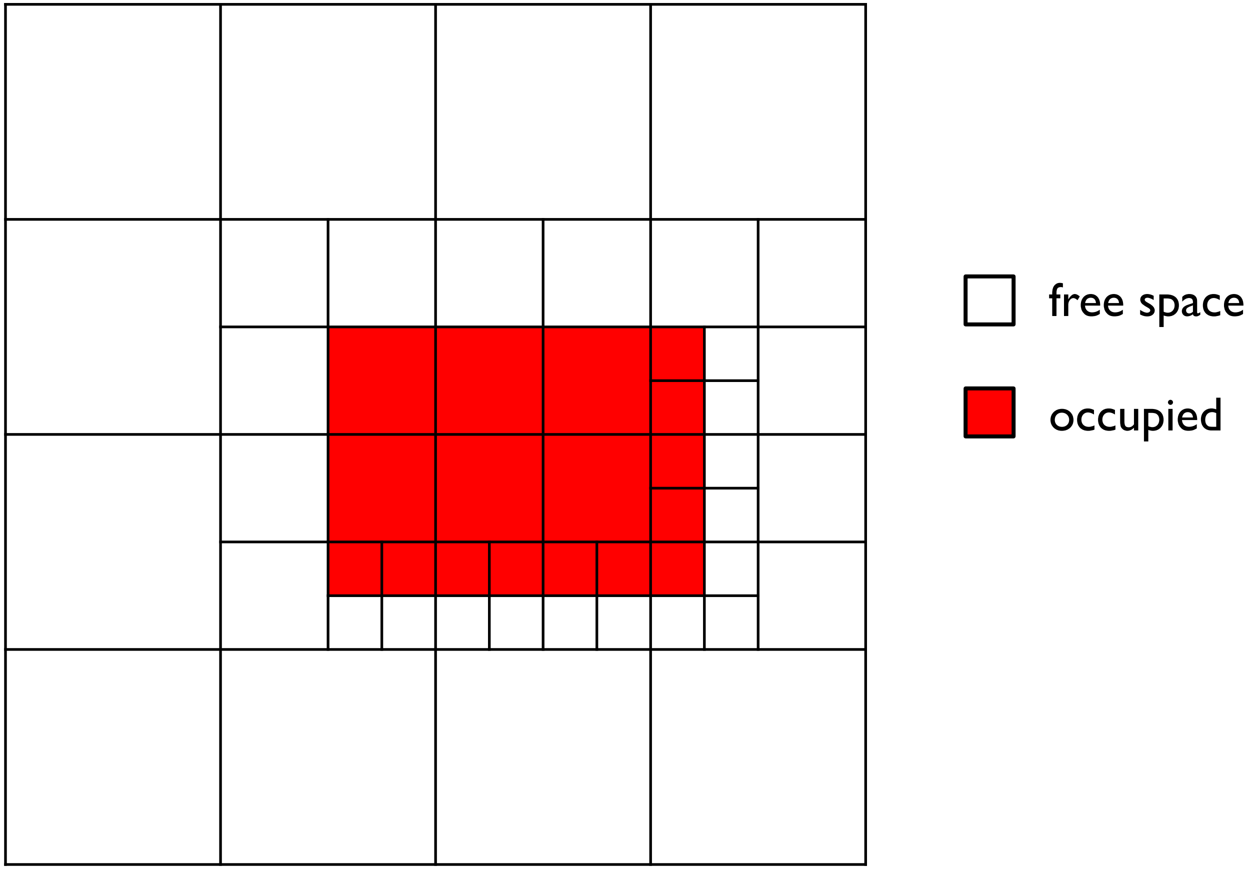
Quadtree Decomposition



Quadtree Decomposition



Quadtree Decomposition

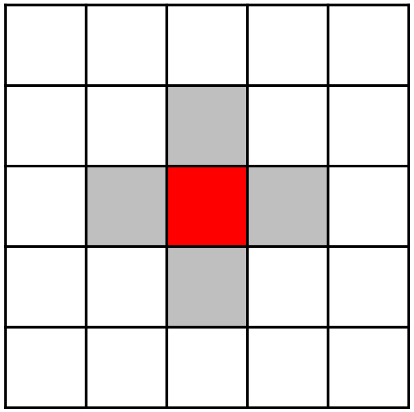


Quadtree Decomposition

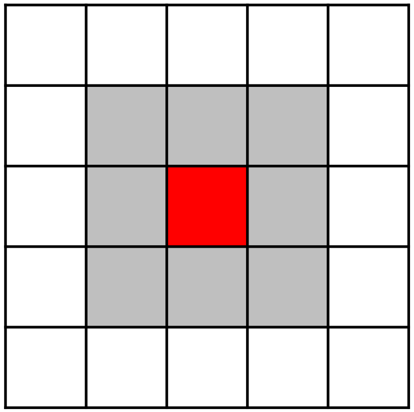
- worst case performance
  - same as uniform subdivision
- if most of the space is occupied or freespace then the representation is compact
- generalizes to N dimensions
- representation changes dramatically if objects move even a small amount

Connectivity in Discrete Sampled Space

- a path on a discrete grid is a sequence of moves between connected cells
- for a square tiling there are two possible definitions of connectivity



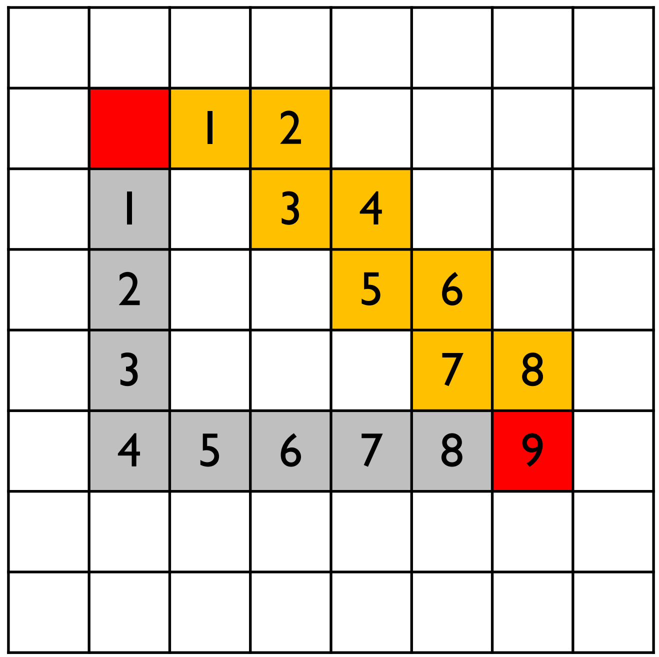
4-connectivity



8-connectivity

4-Connectivity

- on a 4-connected tiling the distance between two cells is called the taxicab distance, rectilinear distance, L_1 distance, L_1 norm, city block distance, or Manhattan distance



Wave-Front Planner

- the wave-front planner finds a path between a start and goal point in spaces represented as a grid where
  - free space is labeled with a 0
  - obstacles are labeled with a 1
  - the goal is labeled with a 2
  - the start is known

	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	2
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
goal	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Wave-Front Planner

- starting at the goal cell

```
L := 2
while start cell is unlabelled
    for each cell C with label L
        for each cell Z connected to C with label 0
            label Z with L+1
    L := L + 1
```

	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3	2
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	3
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
goal	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	0	0	0	0	0	0	0	0	0	0	0	0	0	4	3	2
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	4	3
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
goal	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	0	0	0	0	0	0	0	0	0	0	0	0	5	4	3	2
	0	0	0	0	0	0	0	0	0	0	0	0	0	5	4	3
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
goal	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	0	0	0	14	13	12	11	10	9	8	7	6	5	4	3	2
	0	0	0	0	14	13	12	11	10	9	8	7	6	5	4	3
	0	0	1	1	0	14	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	1	1	1	1	1	1	1	1	1	1
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	1	1	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
goal	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3
	19	18	1	1	15	14	1	1	1	1	1	1	1	1	1	1
	20	19	1	1	16	15	1	1	1	1	1	1	1	1	1	1
	0	20	1	1	17	16	17	18	19	20	0	0	1	1	0	0
	1	1	1	1	18	17	18	19	20	0	0	0	1	1	0	0
	1	1	1	1	19	18	19	20	0	0	0	0	1	1	0	0
	0	0	1	1	20	19	20	0	0	0	0	0	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	0	0	1	1	1	1	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	0	0	0	0	0	0	0	0	0	0	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	0	0
	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
goal	*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3
	19	18	1	1	15	14	1	1	1	1	1	1	1	1	1	1
	20	19	1	1	16	15	1	1	1	1	1	1	1	1	1	1
	21	20	1	1	17	16	17	18	19	20	21	22	1	1	37	38
	1	1	1	1	18	17	18	19	20	21	22	23	1	1	36	37
	1	1	1	1	19	18	19	20	21	22	23	24	1	1	35	36
	0	0	1	1	20	19	20	21	22	23	24	25	1	1	34	35
	0	0	1	1	1	1	1	1	23	24	1	1	1	1	33	34
	0	0	1	1	1	1	1	1	24	25	1	1	1	1	32	33
	0	0	1	1	29	28	27	26	25	26	27	28	29	30	31	32
	0	0	1	1	30	29	28	27	26	27	28	29	30	31	32	33
	0	0	1	1	1	1	1	1	1	1	1	1	1	1	33	34
	0	50	1	1	1	1	1	1	1	1	1	1	1	1	34	35
	50	49	48	46	45	44	43	42	41	40	39	38	37	36	35	36
goal	*	50	49	47	46	45	44	43	42	41	40	39	38	37	36	37

Wave-Front Planner

- to generate a path starting from the start point

```
L := start point label
while not at the goal
   move to any connected cell with label L-1
   L := L-1
```

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3
	19	18	1	1	15	14	1	1	1	1	1	1	1	1	1	1
	20	19	1	1	16	15	1	1	1	1	1	1	1	1	1	1
	21	20	1	1	17	16	17	18	19	20	21	22	1	1	37	38
	1	1	1	1	18	17	18	19	20	21	22	23	1	1	36	37
	1	1	1	1	19	18	19	20	21	22	23	24	1	1	35	36
	0	0	1	1	20	19	20	21	22	23	24	25	1	1	34	35
	0	0	1	1	1	1	1	1	23	24	1	1	1	1	33	34
	0	0	1	1	1	1	1	1	24	25	1	1	1	1	32	33
	0	0	1	1	29	28	27	26	25	26	27	28	29	30	31	32
	0	0	1	1	30	29	28	27	26	27	28	29	30	31	32	33
	0	51	1	1	1	1	1	1	1	1	1	1	1	1	33	34
	51	50	1	1	1	1	1	1	1	1	1	1	1	1	34	35
	50	49	48	46	45	44	43	42	41	40	39	38	37	36	35	36
goal	51	50	49	47	46	45	44	43	42	41	40	39	38	37	36	37

Another Example

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	3	2	3	0	0	0	0	0	0
0	0	0	0	0	0	0	0	3	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	4	3	4	0	0	0	0	0	0
0	0	0	0	0	0	4	3	2	3	4	0	0	0	0	0
0	0	0	0	0	0	0	0	3	4	0	0	0	0	0	0
0	0	0	0	0	0	0	0	4	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

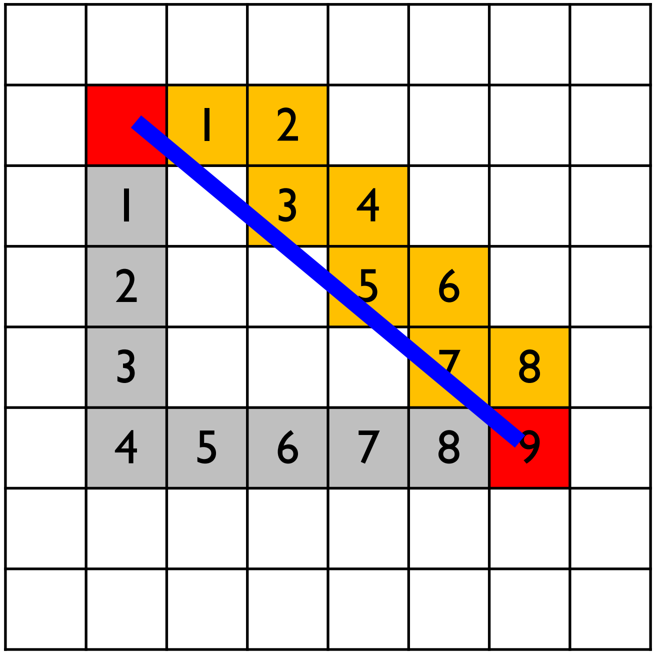
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0
0	0	0	0	0	0	0	7	6	7	0	0	0	0	0	0
0	0	0	0	0	0	7	6	5	6	7	0	0	0	0	0
0	0	0	0	0	7	6	5	4	5	6	7	0	0	0	0
0	0	0	0	7	6	5	4	3	4	5	6	7	0	0	0
0	0	0	7	6	5	4	3	2	3	4	5	6	7	0	0
0	0	0	0	7	6	5	4	3	4	5	6	7	0	0	0
0	0	0	0	0	7	6	5	4	5	6	7	0	0	0	0
0	0	0	0	0	0	7	6	5	6	7	0	0	0	0	0
0	0	0	0	0	0	0	7	6	7	0	0	0	0	0	0
0	0	0	0	0	0	0	0	7	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
*	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0

Wave-Front Planner

- advantage:
  - will find a shortest path (in terms of connectivity) between start and goal if a path exists
  - generalizes to higher dimensions
- disadvantages:
  - path often runs adjacent to obstacles
  - planner searches the entire space with radius R around the goal (where R is the distance between the start and goal)
  - paths restricted by grid connectivity are longer than necessary

Wave-Front Planner

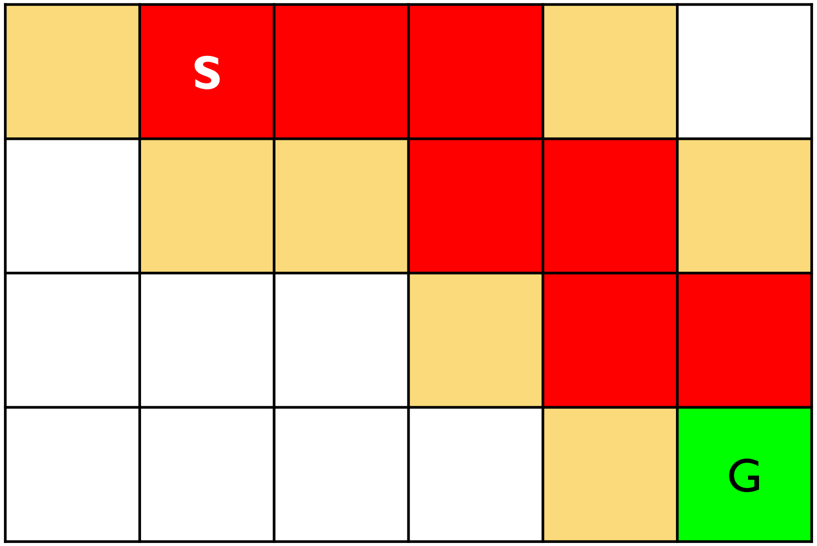
- paths restricted by grid connectivity are longer than necessary
  - Manhattan distance = 9
  - straight line distance = sqrt(16 + 25) = 6.403…



 3/21/2018

Greedy Best-First Search

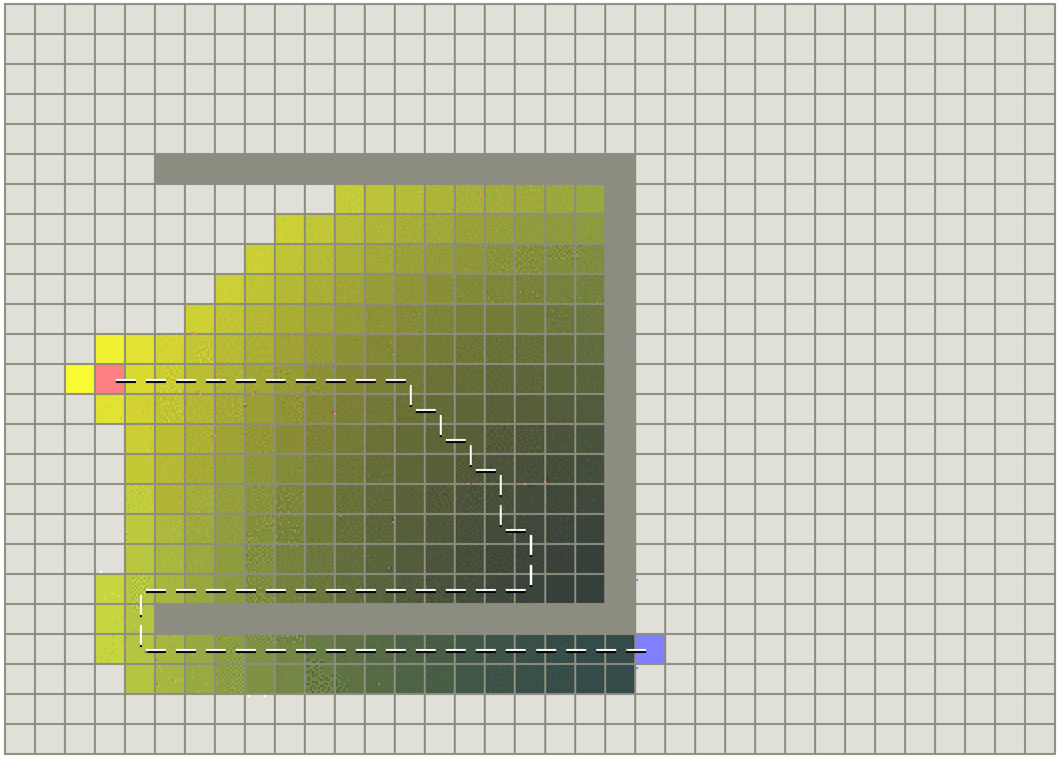
- to avoid searching in all possible directions we might consider searching first in a direction towards the goal
- idea
  - use an estimate (called the heuristic) of how far a cell is from the goal
  - consider the cell whose heuristic distance is the smallest first



possible path with heuristic distance = Euclidean distance

Greedy Best-First Search

- produces expensive paths when there are concave obstacles



A*

- A* is a common algorithm in game AI programming and robotics
  - first described in 1968
  - http://theory.stanford.edu/~amitp/GameProgramming/
- A* is the foundation for Theta*
  - Daniel, Nash, Koenig. Theta*: Any-Angle Planning on Grids, Journal of Artificial Intelligence Research, 39, 2010.
  - path planning on a grid where paths are allowed to pass through cells at any angle (not just using 4- or 8-connectivity)

A*

- A* combines two pieces of information
  - $g(n)$: the cost of the path from the starting point to n
  - $h(n)$: the heuristic cost of the path from n to the goal
  - considers the cell n with the lowest cost

    $$f(n) = g(n) + h(n)$$

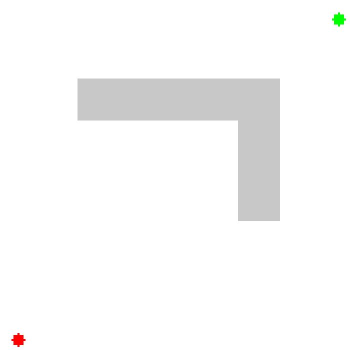
    first
- compromise between Dijkstra's algorithm and greedy best-first search

A*

- the heuristic distance function h(n) affects how the algorithm performs the search
  - h(n) = 0
    - equivalent to Dijkstra's algorithm
  - h(n) <= true cost of moving from n to the goal
    - guaranteed to find a shortest path
    - the smaller h(n) the more it expands the search to cells closer to the start
  - h(n) = true cost of moving from n to the goal
    - will find a best path with the minimal amount of searching
  - h(n) > true cost of moving from n to the goal some of the time
    - not guaranteed to find a shortest path but might find a path in a shorter amount of time
  - h(n) >> true cost of moving from n to the goal
    - behaves like greedy best-first search

A*

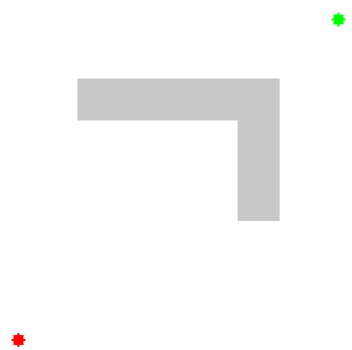
- h(n) <= true cost of moving from n to the goal



http://en.wikipedia.org/wiki/File:Astar_progress_animation.gif

A*

- h(n) > true cost of moving from n to the goal



http://en.wikipedia.org/wiki/File:Weighted_A_star_with_eps_5.gif

Potential Functions

- in continuous space potential functions can be used for path planning
- a potential function is a differentiable real-valued function U

$$U : \mathrm{R}^m \to \mathrm{R}$$

  - i.e., U assigns a scalar real value to every point in space
- potential functions you might know
  - gravitational potential
  - electrostatic potential

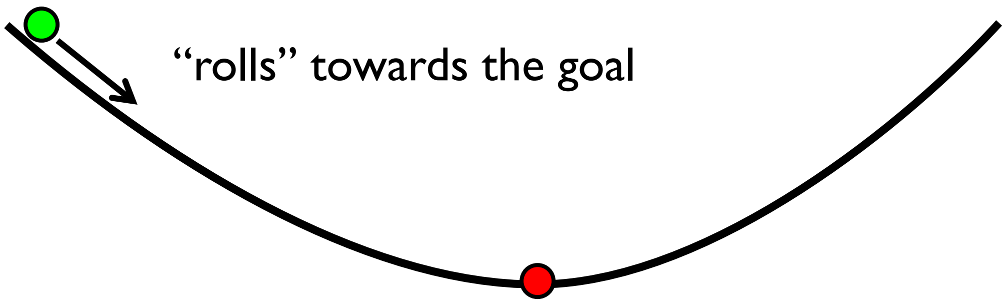
Goal Potential

- the goal potential should be an attractive potential
  - small near the goal
  - large far from the goal
  - monotonically increasing
    - nice too if it is continuously differentiable

Goal Potential

- consider the quadratic potential

$$U_{\text{attract}} = \alpha \left\| q - q_{\text{goal}} \right\|^2$$



Goal Potential

- "rolling towards the goal" can be accomplished using gradient descent

$$
\begin{aligned}
F &= \nabla U_{\text{attract}} \\
&= \begin{bmatrix} \partial U / \partial x \\ \partial U / \partial y \end{bmatrix} \\
&= \alpha\left(q - q_{\text{goal}}\right)
\end{aligned}
$$

- gradient descent
  - starting at initial configuration, take a small step in the direction opposite to the gradient F until $|F| = 0$

Goal Potential

- notice that the wave-front planner basically works this way
  - it defines a potential where there is only one minimum
    - the minimum is located at the goal
  - it then uses gradient descent to move towards the goal

	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2
	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3
	19	18	1	1	15	14	1	1	1	1	1	1	1	1	1	1
	20	19	1	1	16	15	1	1	1	1	1	1	1	1	1	1
	21	20	1	1	17	16	17	18	19	20	21	22	1	1	37	38
	1	1	1	1	18	17	18	19	20	21	22	23	1	1	36	37
	1	1	1	1	19	18	19	20	21	22	23	24	1	1	35	36
	0	0	1	1	20	19	20	21	22	23	24	25	1	1	34	35
	0	0	1	1	1	1	1	1	23	24	1	1	1	1	33	34
	0	0	1	1	1	1	1	1	24	25	1	1	1	1	32	33
	0	0	1	1	29	28	27	26	25	26	27	28	29	30	31	32
	0	0	1	1	30	29	28	27	26	27	28	29	30	31	32	33
	0	51	1	1	1	1	1	1	1	1	1	1	1	1	33	34
	51	50	1	1	1	1	1	1	1	1	1	1	1	1	34	35
	50	49	48	46	45	44	43	42	41	40	39	38	37	36	35	36
start	51	50	49	47	46	45	44	43	42	41	40	39	38	37	36	37