Potential Functions

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

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

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

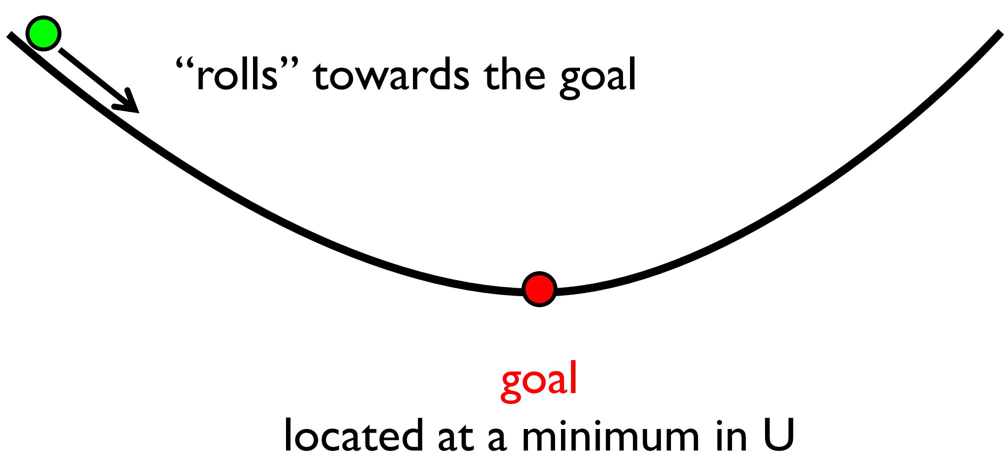
Goal Potential

- 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

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



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

Day 26

Potential Functions (cont)

Obstacle Potential

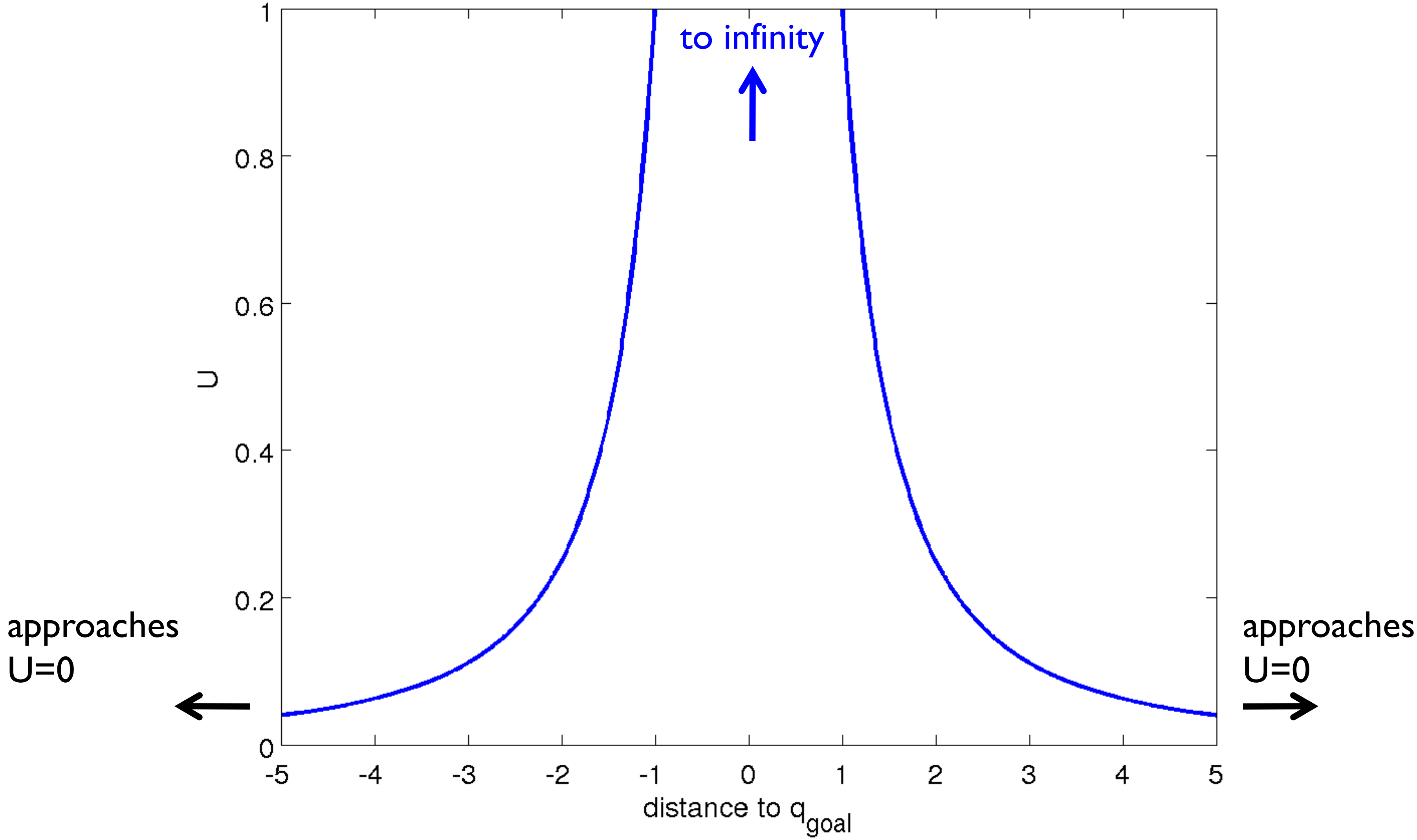
- obstacles should have a repulsive potential to keep the robot away from the obstacle
  - the repulsive force should increase closer to the obstacle
  - often modeled as a potential barrier that rises to infinity as the robot approaches the obstacle; for example

$$U_{\text{obstacle}} = \beta \frac{1}{\| q - q_{\text{obstacle}} \|^2}$$

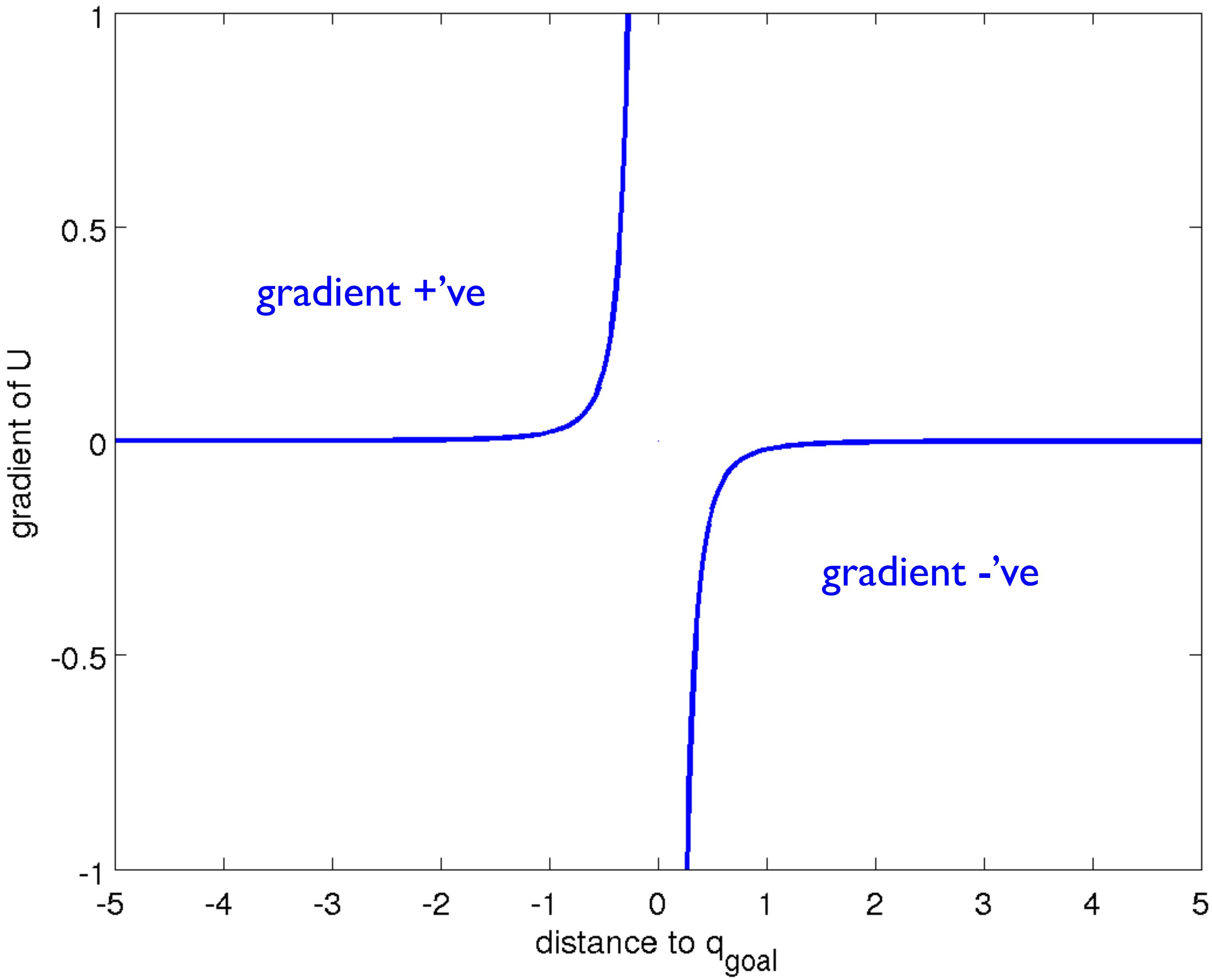
  - where q_{obstacle} is the closest point on the obstacle

Obstacle Potential

$$U_{\text{obstacle}} = \beta \frac{1}{\| q - q_{\text{obstacle}} \|^2}$$

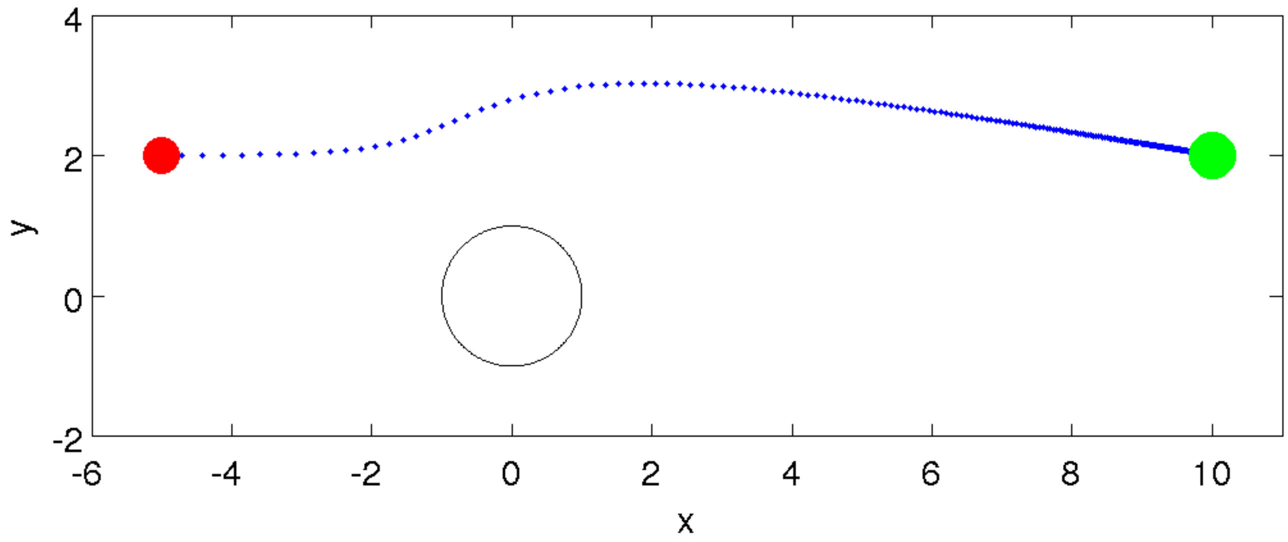


Obstacle Gradient



Obstacle Potential

- one problem with the previous obstacle potential
  - potential (and gradient) is always non-zero away from the obstacle
    - an obstacle far away from the robot will influence the path



Obstacle Potential

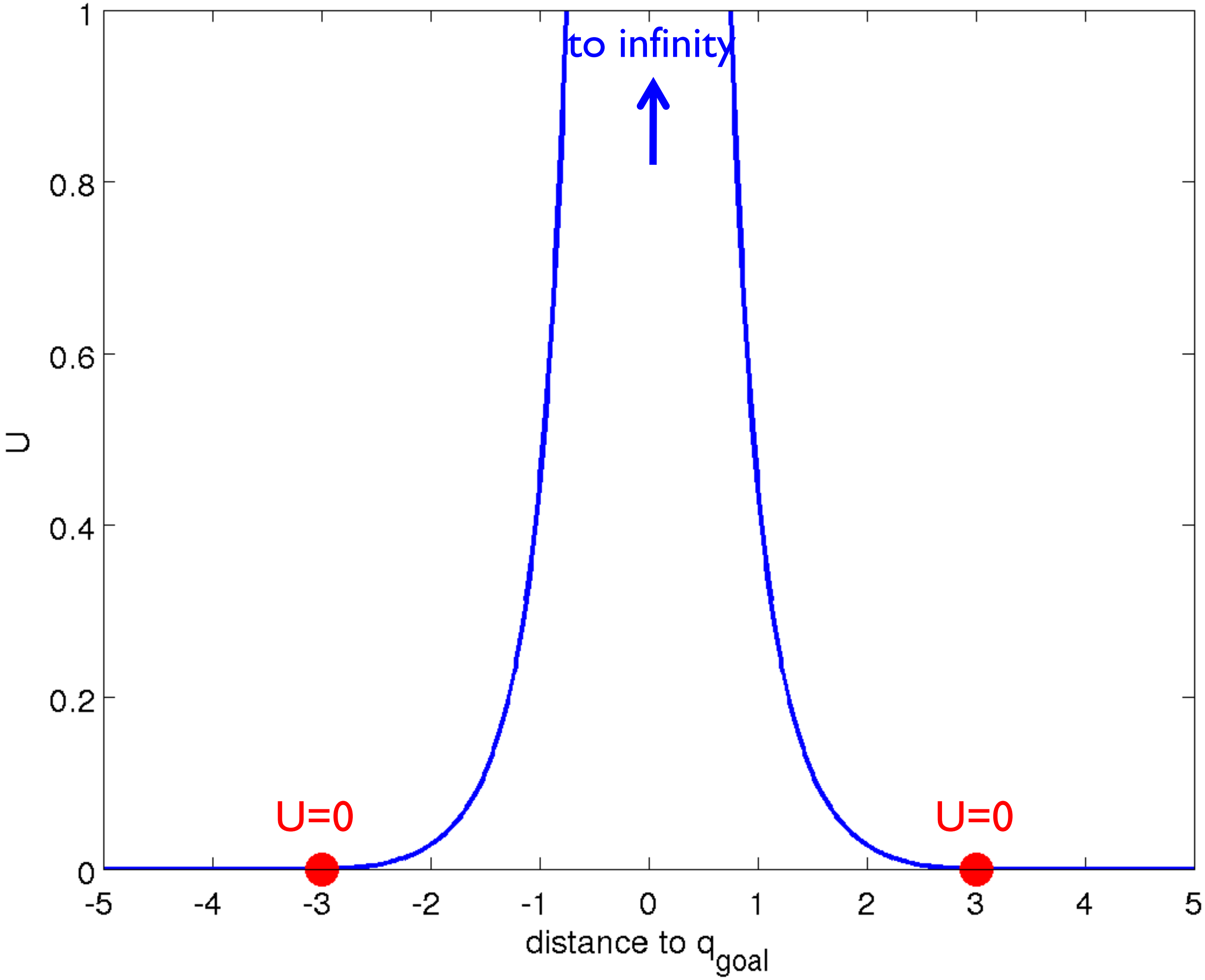
- possible solution
  - choose a potential function that equals zero a distance Q^* away from the obstacle

$$U_{\text{obstacle}} = \begin{cases} \beta\left(\dfrac{1}{D(q)} - \dfrac{1}{Q^*}\right)^2, & D(q) \le Q^* \\ 0, & D(q) > Q^* \end{cases}$$

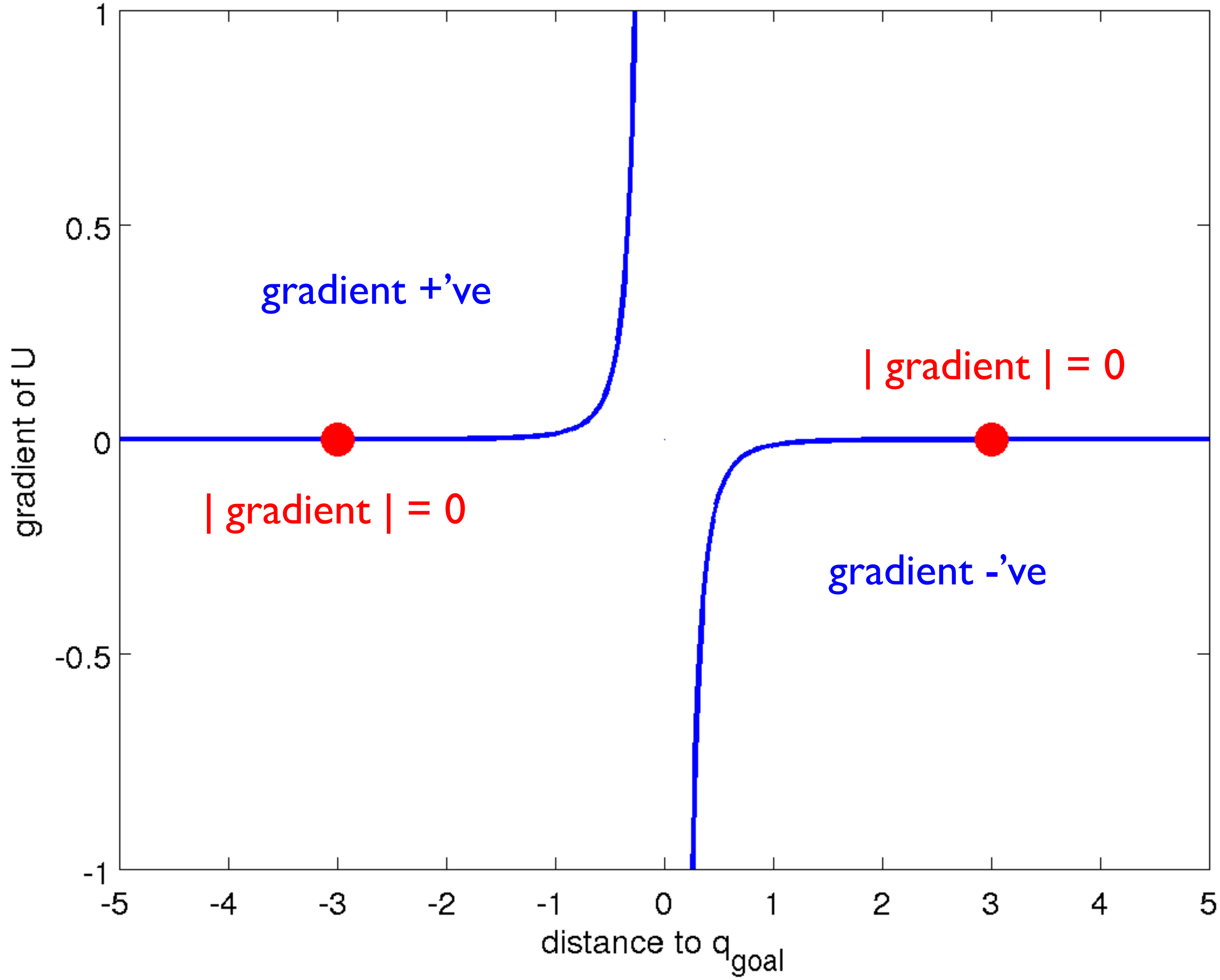
  - where $D(q)$ is the distance between the robot and the obstacle

Obstacle Potential

$$U_{\text{obstacle}} = \begin{cases} \beta\left(\dfrac{1}{D(q)} - \dfrac{1}{Q^*}\right)^2, & D(q) \le Q^* \\ 0, & D(q) > Q^* \end{cases}$$



Obstacle Gradient



Total Potential

- the total potential field is simply the sum of the attractive and repulsive potentials

$$U(q) = U_{\text{goal}}(q) + U_{\text{obstacle}}(q)$$

  - where q is the location of the robot

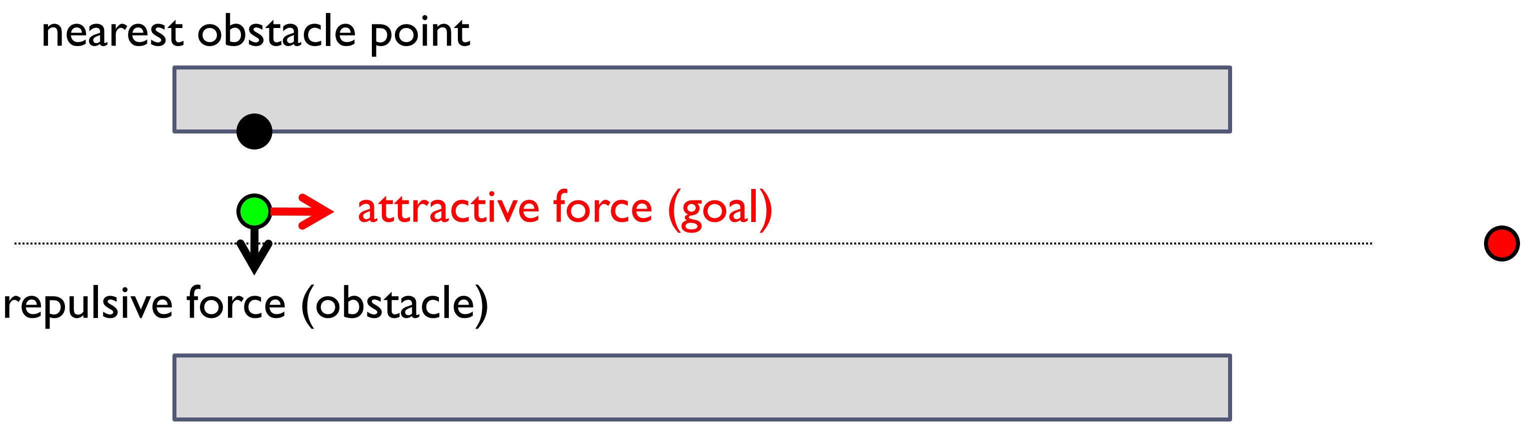
Multiple Obstacles

- one approach for multiple obstacles is to consider only the nearest obstacle

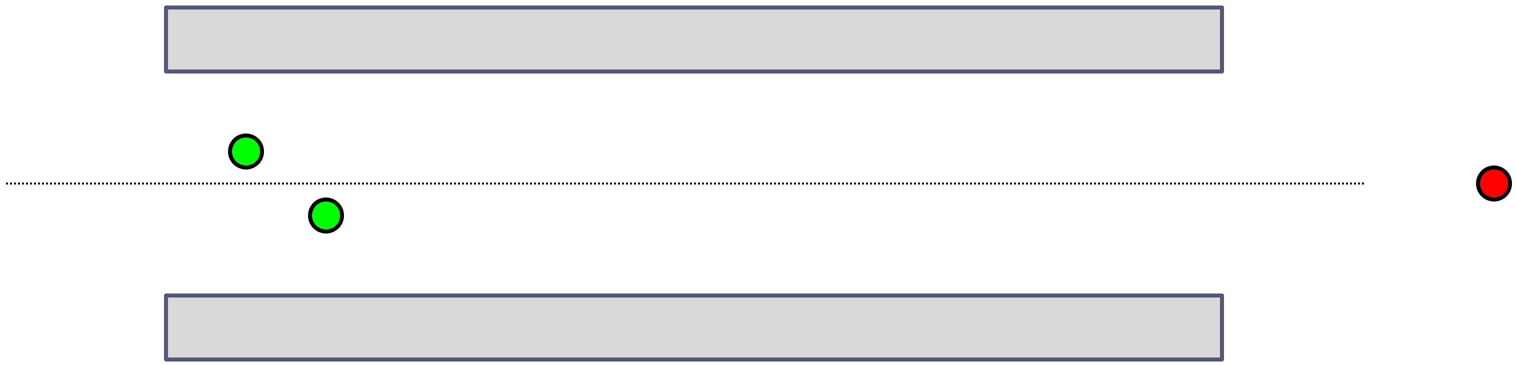
$$U(q) = U_{\text{goal}}(q) + U_{\text{nearest obstacle}}(q)$$

  - this can lead to oscillating paths when the robot is almost equidistant to two or more obstacles

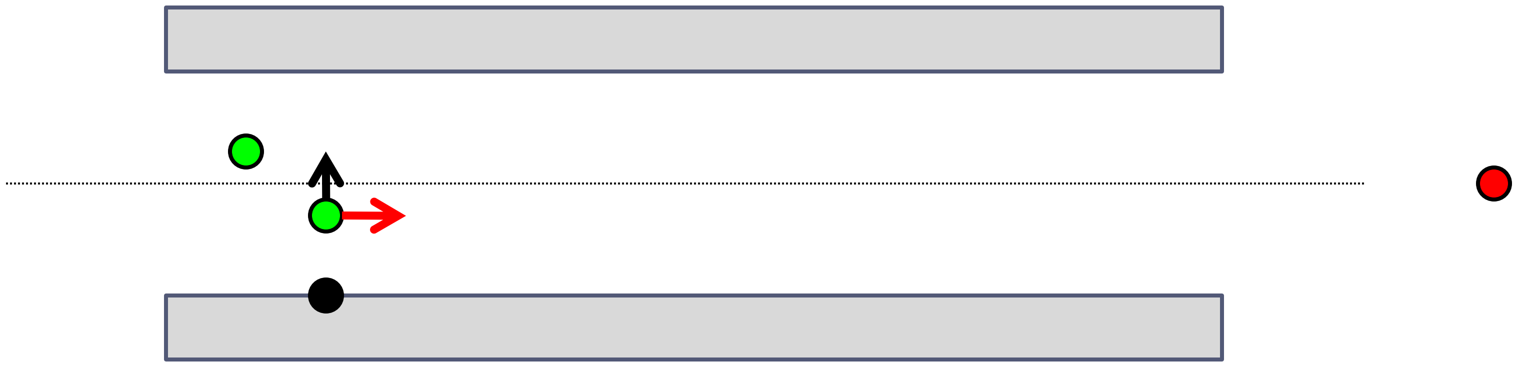
Multiple Obstacles



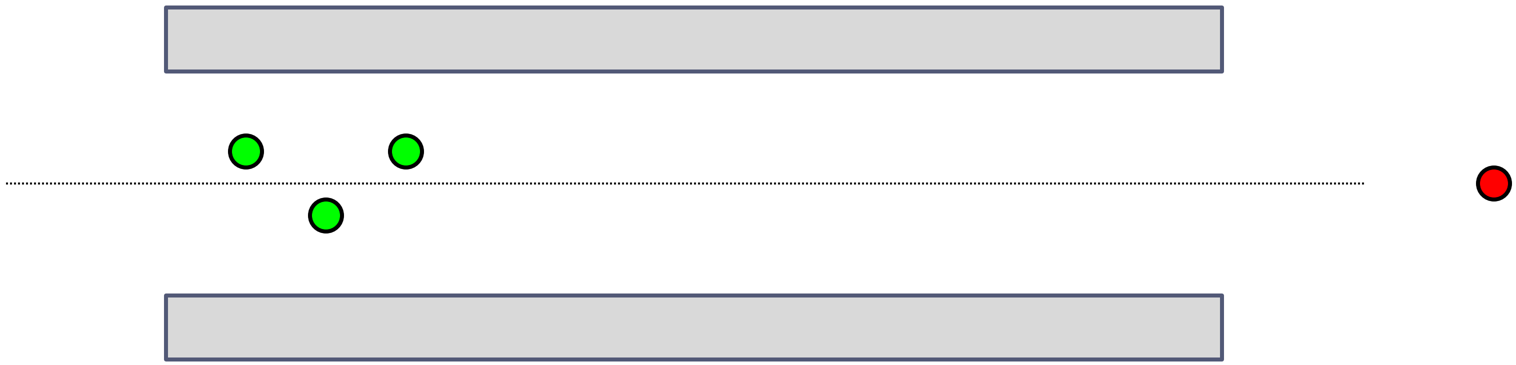
Multiple Obstacles



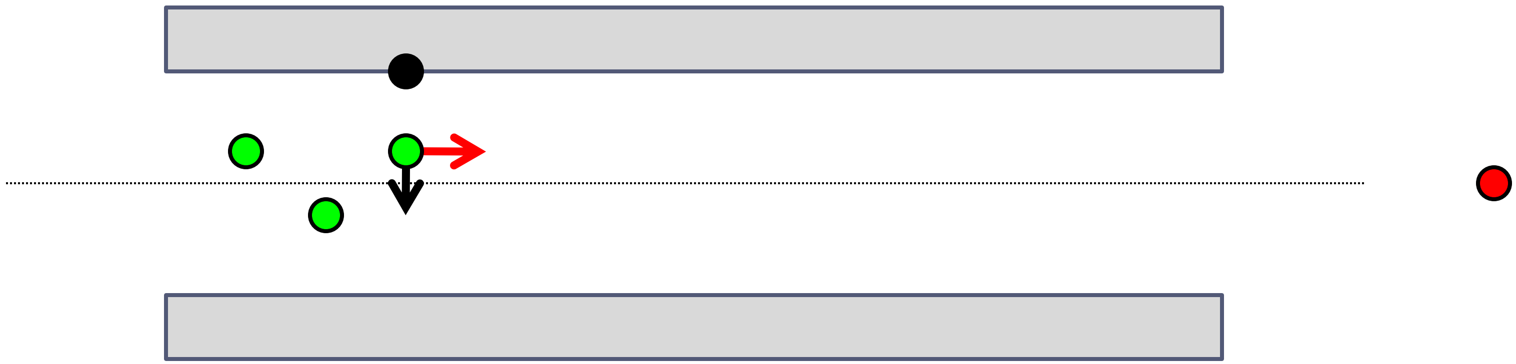
Multiple Obstacles



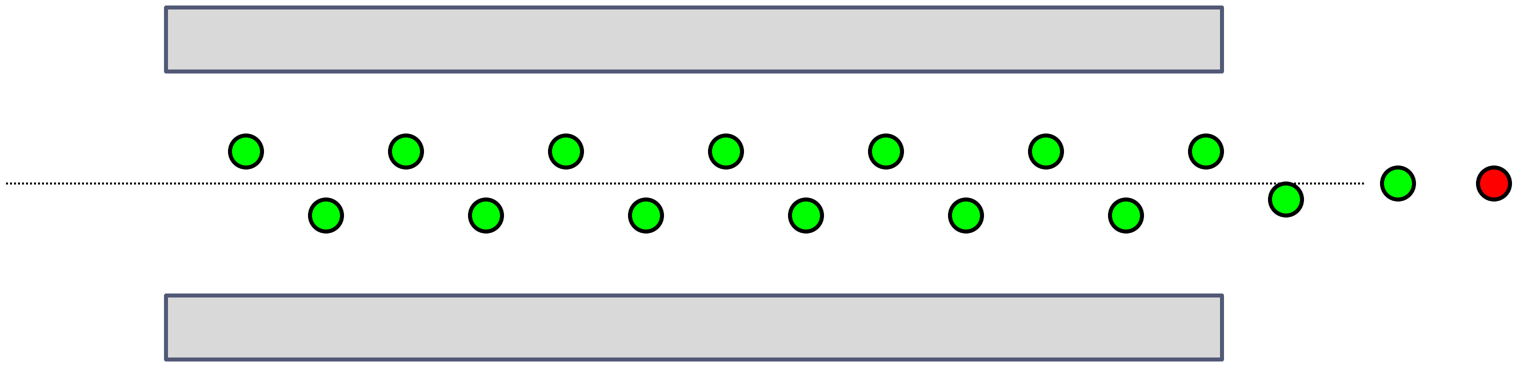
Multiple Obstacles



Multiple Obstacles



Multiple Obstacles



Multiple Obstacles

- an alternative approach is to consider the contributions to the total potential from all obstacles

$$U(q) = U_{\text{goal}}(q) + \sum_i U_{\text{obstacle},i}(q)$$

  - where $U_{\text{obstacle},\,i}$ is the repulsive potential contribution from the i^{th} obstacle

Computing Distances on a Grid

- the brushfire algorithm can be used to compute distances on a grid where
  - free space is labeled with a 0
  - obstacles are labeled with a 1
- outputs
  - grid labels equal to the distance to the nearest obstacle
    - grid labels can be used to compute gradients
- like the wave-front planner, you need to decide between 4- and 8-connectivity

Brushfire Algorithm

```
for each cell labeled 1
   label each adjacent free-space cell with 2
L := 2
do
   for each cell labeled L
        label each adjacent free-space cell with L+1
   L := L+1
while there are still free-space cells remaining
```

Brushfire Algorithm

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	0	0	0	0	0	0	0	0	1
1	0	0	1	1	1	1	1	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	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	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	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1
1	0	0	1	0	0	0	1	0	0	0	0	0	0	0	1
1	0	0	1	1	1	1	1	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	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	1	1	1	1	1

Brushfire Algorithm

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	2	2	1	2	2	2	2	2	2	2	2	2	2	2	1
1	2	2	1	2	0	0	0	0	0	0	0	0	0	2	1
1	2	2	1	2	2	2	2	0	0	0	0	0	0	2	1
1	2	2	1	1	1	1	1	2	0	0	0	0	0	2	1
1	2	0	2	2	2	2	2	0	0	0	0	0	0	2	1
1	2	0	0	0	0	0	0	0	0	0	0	0	0	2	1
1	2	0	0	0	0	0	0	0	0	0	0	0	0	2	1
1	2	0	0	0	0	0	0	0	0	0	0	0	0	2	1
1	2	0	2	0	0	0	2	0	0	0	0	0	0	2	1
1	2	2	1	2	0	2	1	2	0	0	0	0	0	2	1
1	2	2	1	2	2	2	1	2	0	0	0	0	0	2	1
1	2	2	1	1	1	1	1	2	0	0	0	0	0	2	1
1	2	0	2	2	2	2	2	0	0	0	0	0	0	2	1
1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Brushfire Algorithm

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	2	2	1	2	2	2	2	2	2	2	2	2	2	2	1
1	2	2	1	2	3	3	3	3	3	3	3	3	3	2	1
1	2	2	1	2	2	2	2	3	0	0	0	0	3	2	1
1	2	2	1	1	1	1	1	2	3	0	0	0	3	2	1
1	2	3	2	2	2	2	2	3	0	0	0	0	3	2	1
1	2	3	3	3	3	3	3	0	0	0	0	0	3	2	1
1	2	3	0	0	0	0	0	0	0	0	0	0	3	2	1
1	2	3	3	0	0	0	3	0	0	0	0	0	3	2	1
1	2	3	2	3	0	3	2	3	0	0	0	0	3	2	1
1	2	2	1	2	3	2	1	2	3	0	0	0	3	2	1
1	2	2	1	2	2	2	1	2	3	0	0	0	3	2	1
1	2	2	1	1	1	1	1	2	3	0	0	0	3	2	1
1	2	3	2	2	2	2	2	3	3	3	3	3	3	2	1
1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Brushfire Algorithm

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	2	2	1	2	2	2	2	2	2	2	2	2	2	2	1
1	2	2	1	2	3	3	3	3	3	3	3	3	3	2	1
1	2	2	1	2	2	2	2	3	4	4	4	4	3	2	1
1	2	2	1	1	1	1	1	2	3	4	0	4	3	2	1
1	2	3	2	2	2	2	2	3	4	0	0	4	3	2	1
1	2	3	3	3	3	3	3	4	0	0	0	4	3	2	1
1	2	3	4	4	4	4	4	0	0	0	0	4	3	2	1
1	2	3	3	4	0	4	3	4	0	0	0	4	3	2	1
1	2	3	2	3	4	3	2	3	4	0	0	4	3	2	1
1	2	2	1	2	3	2	1	2	3	4	0	4	3	2	1
1	2	2	1	2	2	2	1	2	3	4	0	4	3	2	1
1	2	2	1	1	1	1	1	2	3	4	4	4	3	2	1
1	2	3	2	2	2	2	2	3	3	3	3	3	3	2	1
1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Brushfire Algorithm

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	2	2	1	2	2	2	2	2	2	2	2	2	2	2	1
1	2	2	1	2	3	3	3	3	3	3	3	3	3	2	1
1	2	2	1	2	2	2	2	3	4	4	4	4	3	2	1
1	2	2	1	1	1	1	1	2	3	4	5	4	3	2	1
1	2	3	2	2	2	2	2	3	4	5	5	4	3	2	1
1	2	3	3	3	3	3	3	4	5	0	5	4	3	2	1
1	2	3	4	4	4	4	4	5	0	0	5	4	3	2	1
1	2	3	3	4	5	4	3	4	5	0	5	4	3	2	1
1	2	3	2	3	4	3	2	3	4	5	5	4	3	2	1
1	2	2	1	2	3	2	1	2	3	4	5	4	3	2	1
1	2	2	1	2	2	2	1	2	3	4	5	4	3	2	1
1	2	2	1	1	1	1	1	2	3	4	4	4	3	2	1
1	2	3	2	2	2	2	2	3	3	3	3	3	3	2	1
1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Brushfire Algorithm

1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	2	2	1	2	2	2	2	2	2	2	2	2	2	2	1
1	2	2	1	2	3	3	3	3	3	3	3	3	3	2	1
1	2	2	1	2	2	2	2	3	4	4	4	4	3	2	1
1	2	2	1	1	1	1	1	2	3	4	5	4	3	2	1
1	2	3	2	2	2	2	2	3	4	5	5	4	3	2	1
1	2	3	3	3	3	3	3	4	5	6	5	4	3	2	1
1	2	3	4	4	4	4	4	5	6	6	5	4	3	2	1
1	2	3	3	4	5	4	3	4	5	6	5	4	3	2	1
1	2	3	2	3	4	3	2	3	4	5	5	4	3	2	1
1	2	2	1	2	3	2	1	2	3	4	5	4	3	2	1
1	2	2	1	2	2	2	1	2	3	4	5	4	3	2	1
1	2	2	1	1	1	1	1	2	3	4	4	4	3	2	1
1	2	3	2	2	2	2	2	3	3	3	3	3	3	2	1
1	2	2	2	2	2	2	2	2	2	2	2	2	2	2	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Brushfire Algorithm

- the gradient of distance at a cell is determined by computing differences with neighboring cells