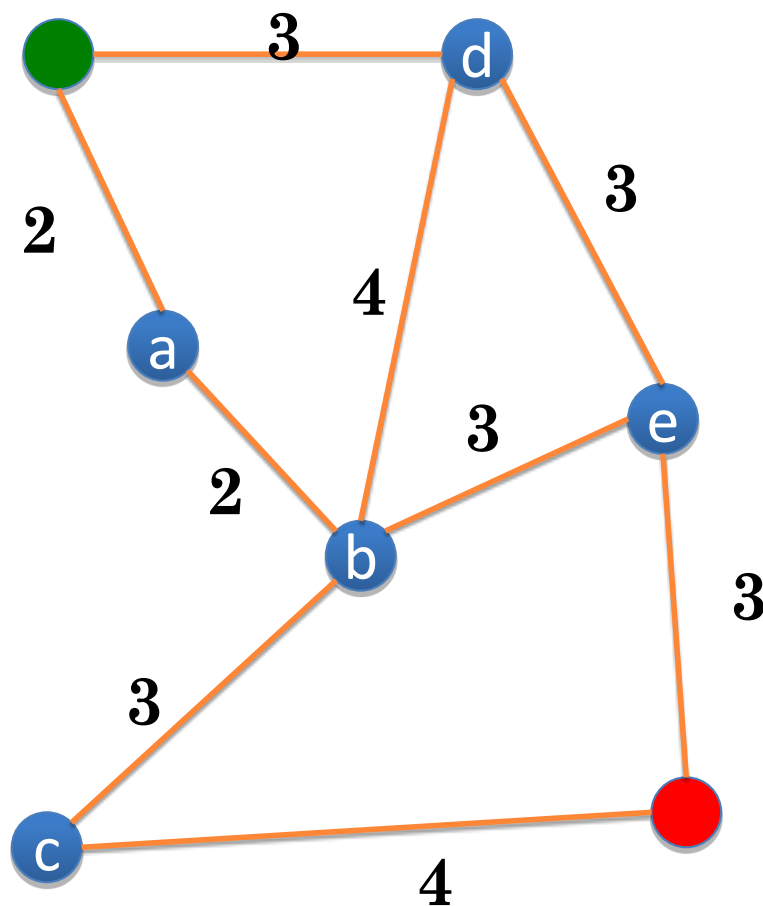




## 4.3 路径搜索算法

# 最优路径搜索算法

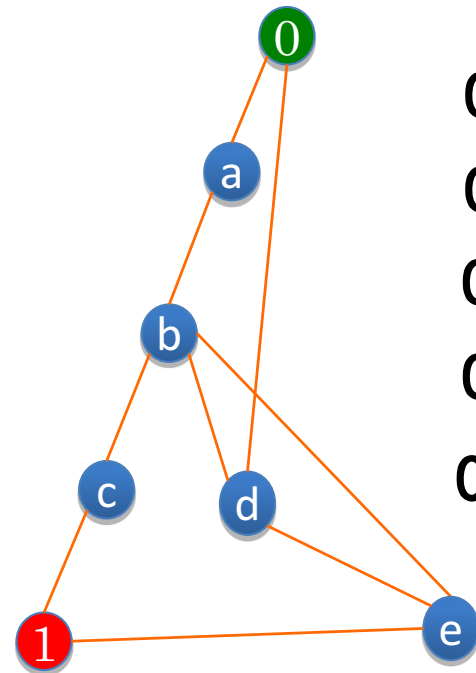
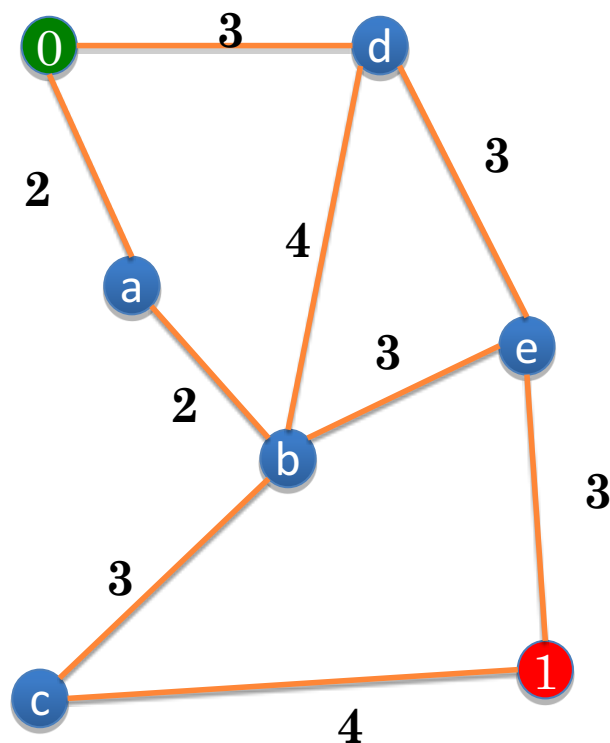
- 在构建形成的连通图中搜索最优路径



# 最优路径搜索算法

## ○ 精确算法：生成精确的最优解

- 深度优先法、广度优先法



0abc1 路径长11

0abde1 路径长14

0abe1 路径长10

0dbc1 路径长14

0de1 路径长9

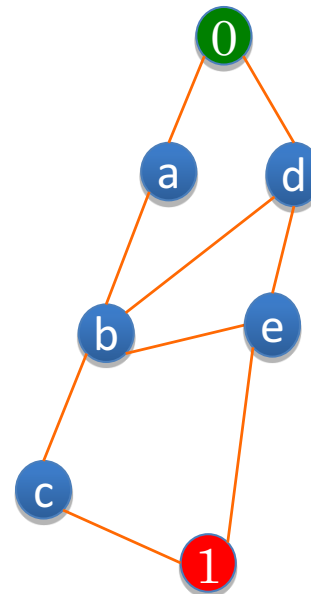
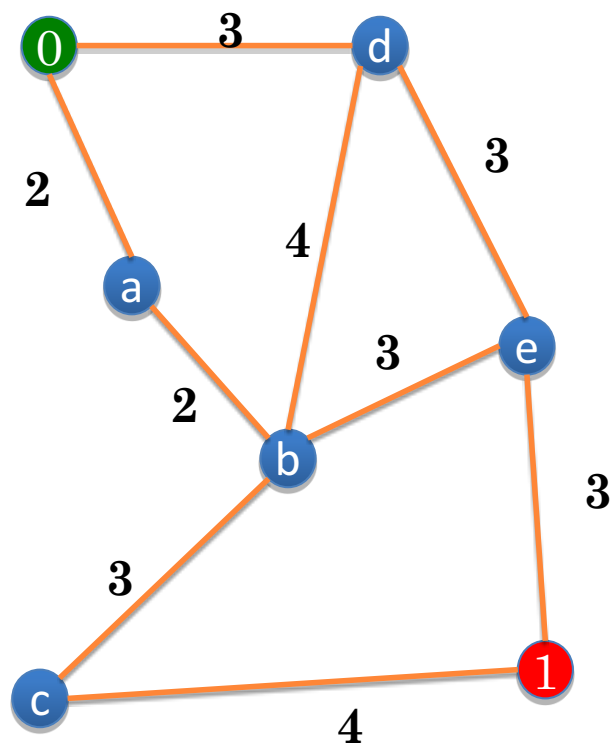
深度优先法



# 最优路径搜索算法

## ○ 精确算法：生成精确的最优解

- 深度优先法、广度优先法



0de1 路径长9

0dbe1 路径长13

0abe1 路径长10

0abc1 路径长11

0dbc1 路径长14

广度优先法

# 最优路径搜索算法

## ○ 精确算法：生成精确的最优解

- 深度优先法、广度优先法
- 广度优先法算法优化形成了Dijkstra算法
- 遍历获得所有路径后选择最优解
- 存在问题：耗时，难以满足机器人在线快速规划要求

## ○ 近似算法

- 启发式搜索算法：A\*, D\*, Focused D\*等
- 准启发式搜索算法：退火、进化和蚁群优化等



# A\*

- 是一种经典启发式搜索算法
- 根据**评估函数**在静态连通图中寻找最优路径
  - 当前搜索结点往下选择下一步结点时，通过启发式函数进行选择，选择代价最少的结点作为下一步搜索结点而跳转其上

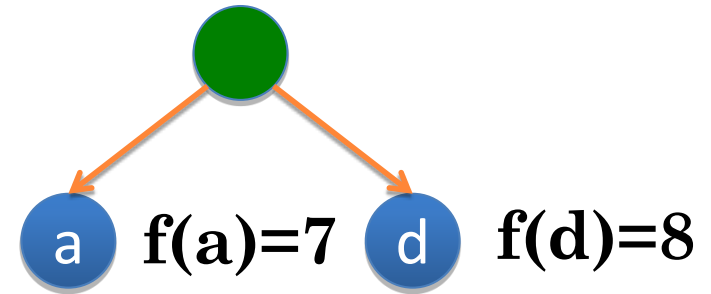
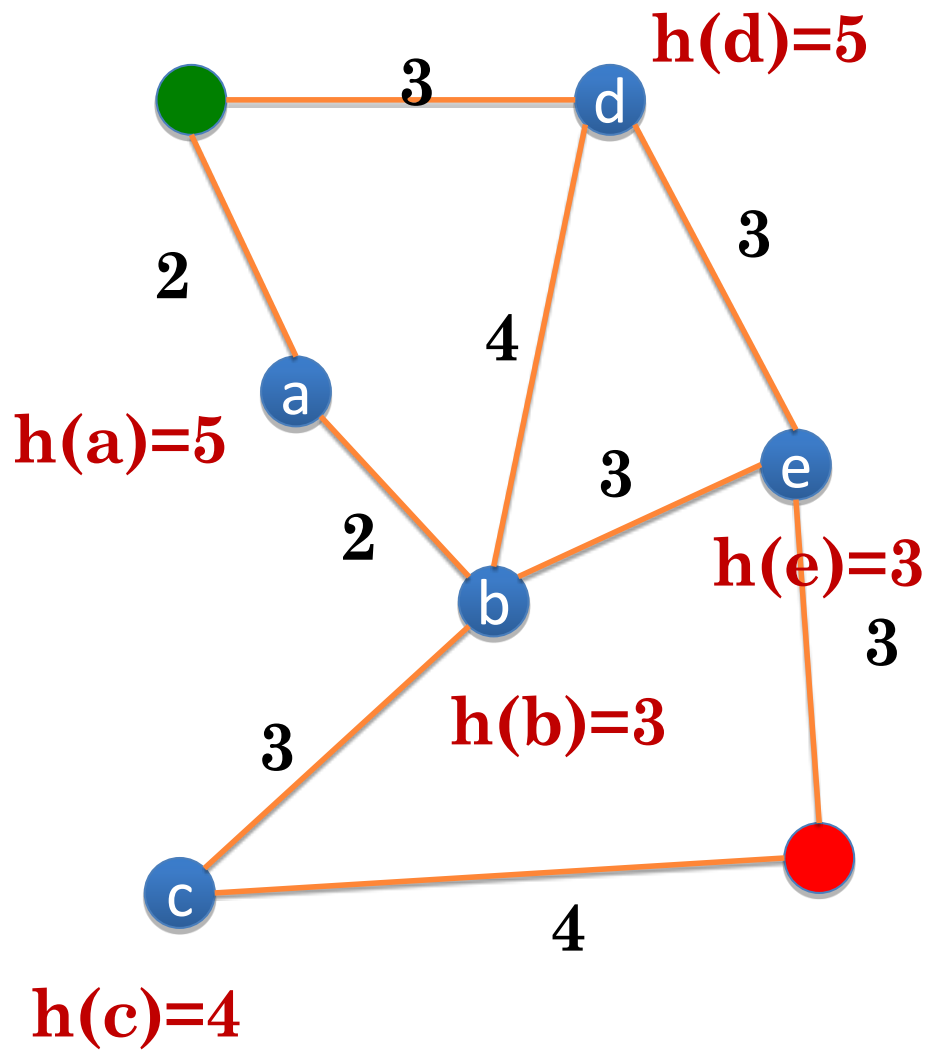
评估函数 
$$f(n) = g(n) + h(n)$$

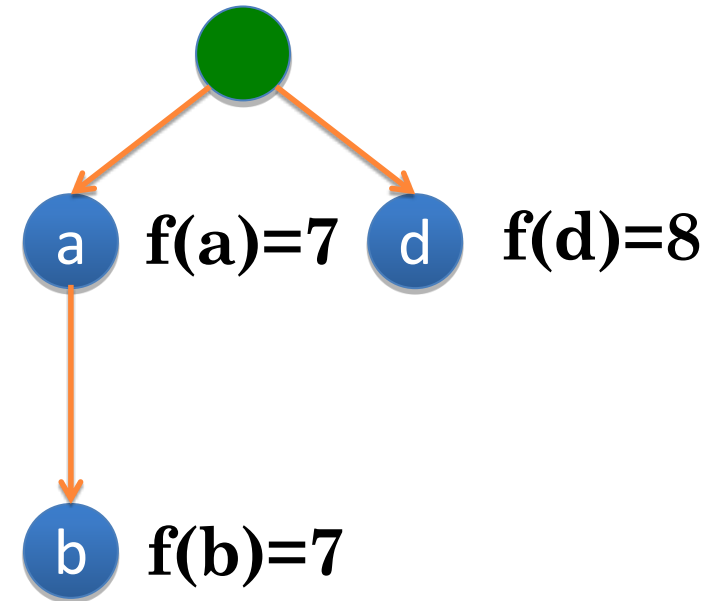
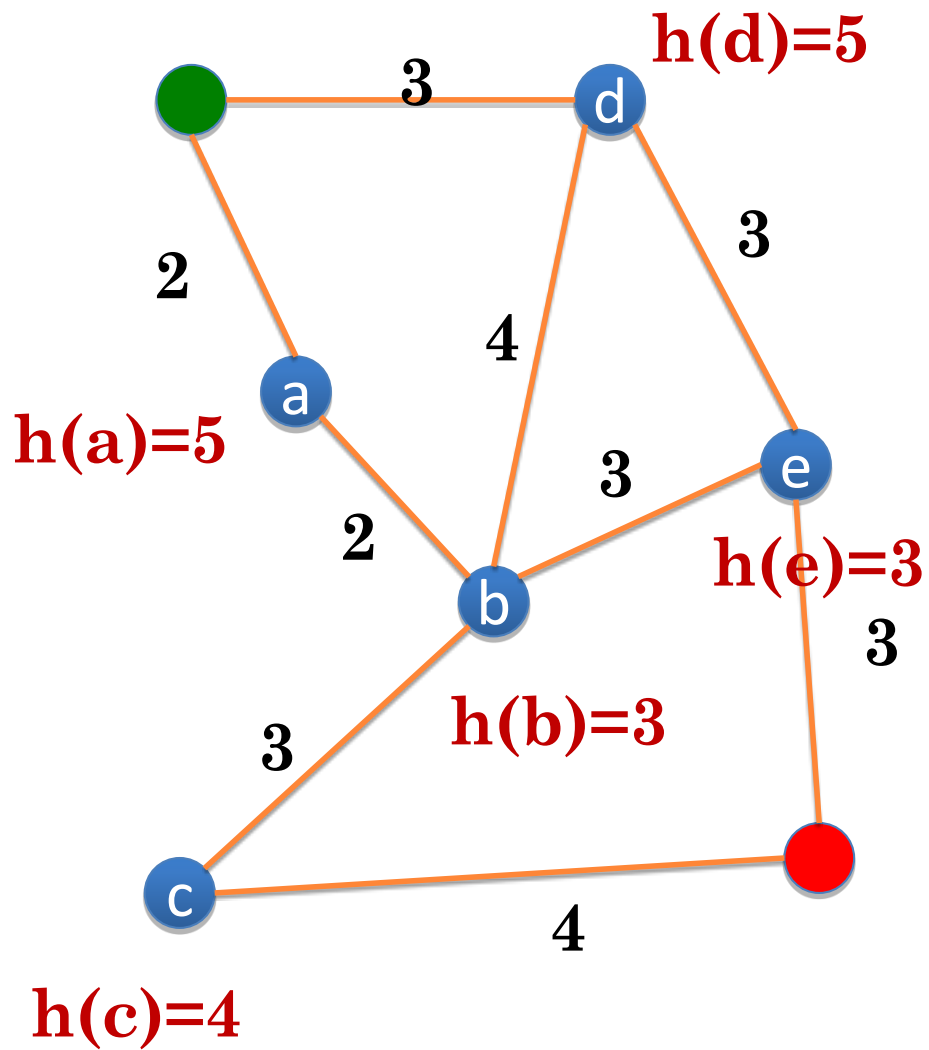
n 表示节点

g(n) 表示从起始点到节点 的实际代价

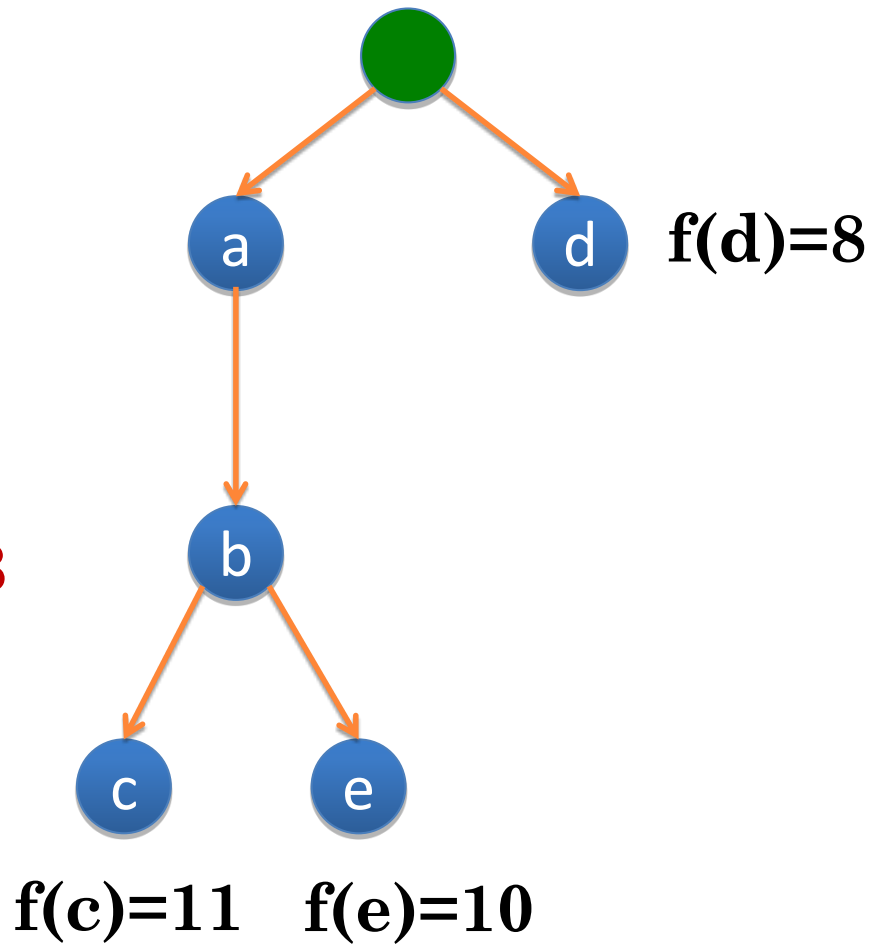
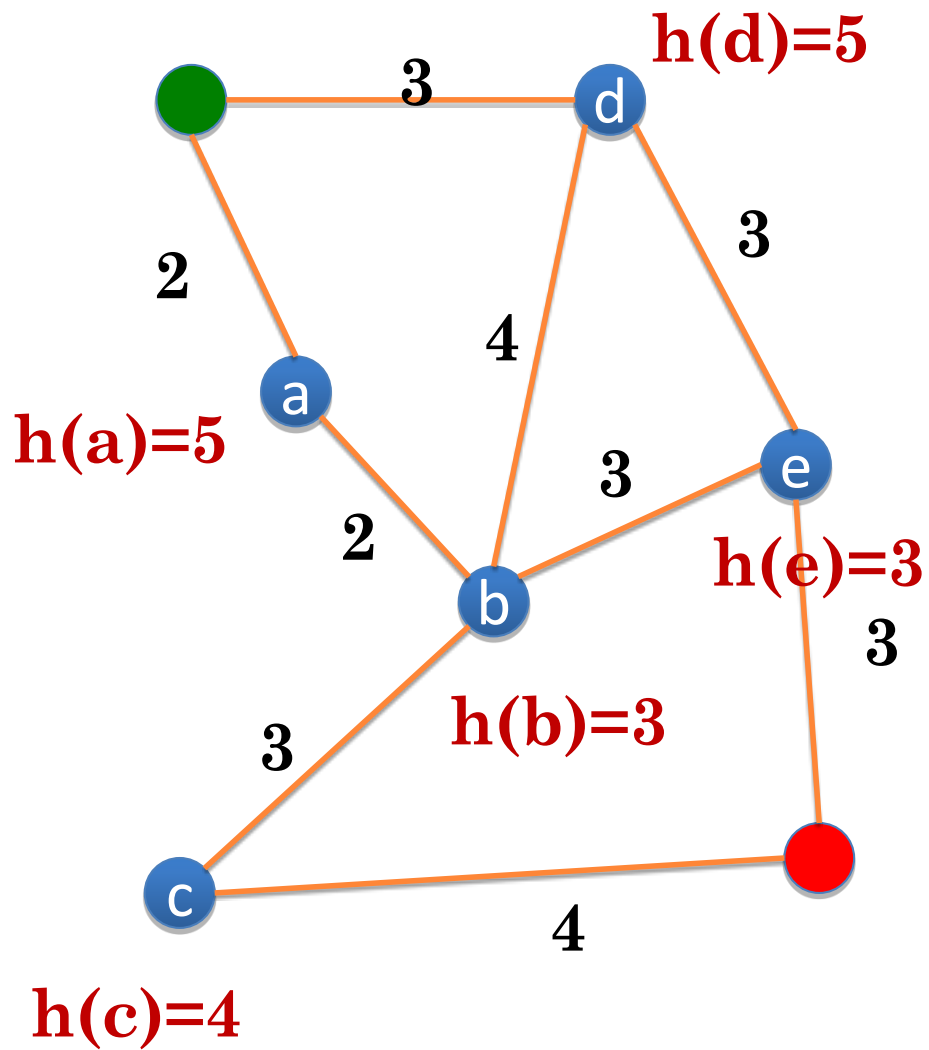
h(n) 为从节点 到目标点的最佳路径的估计代价

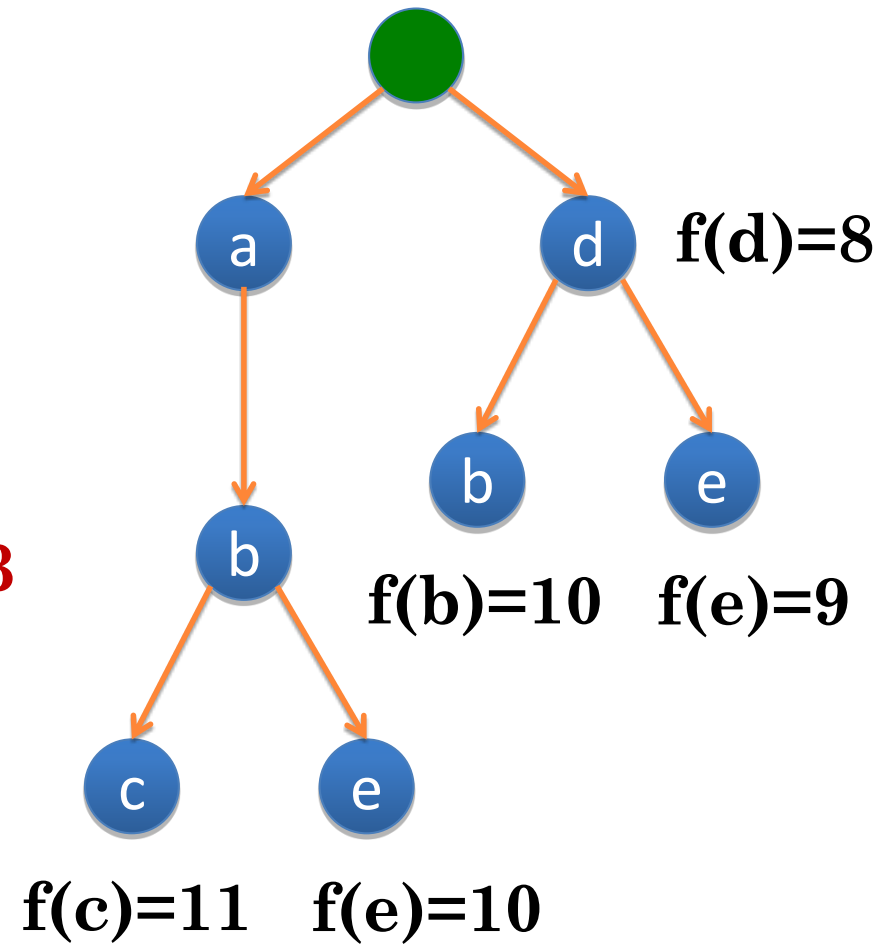
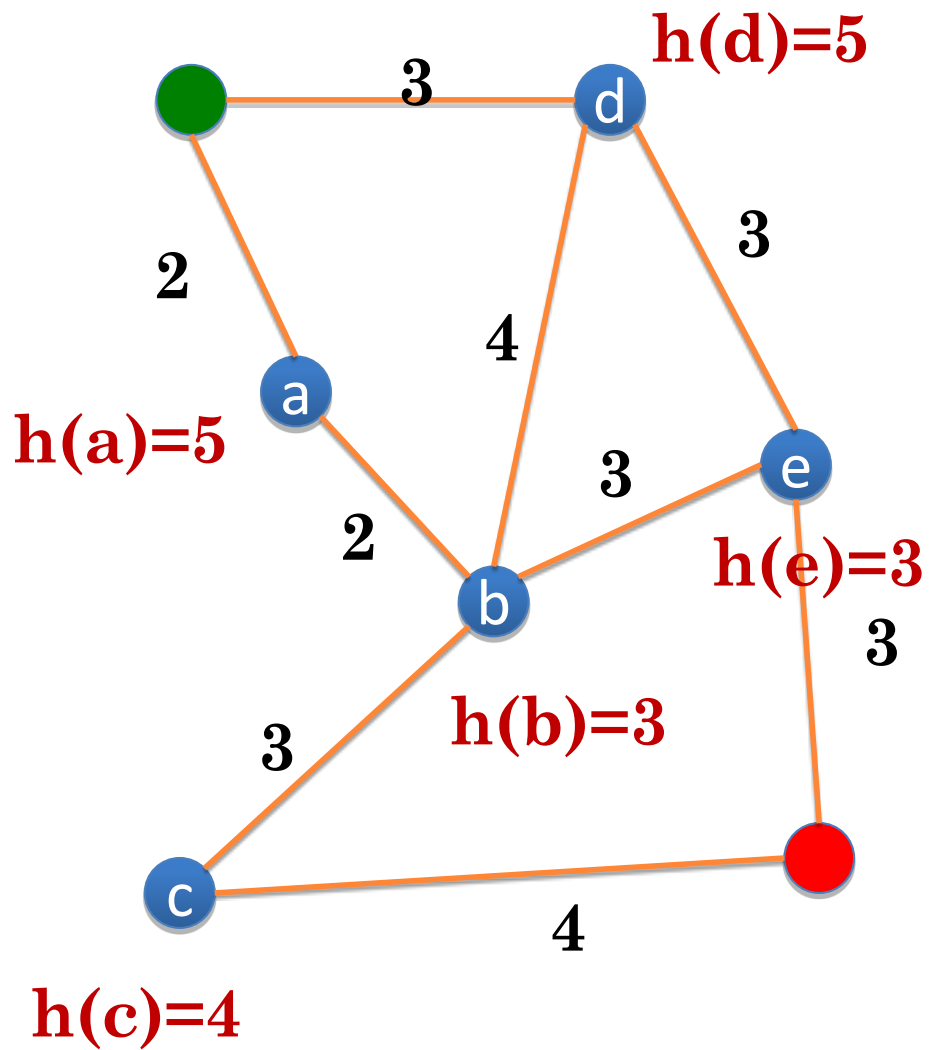


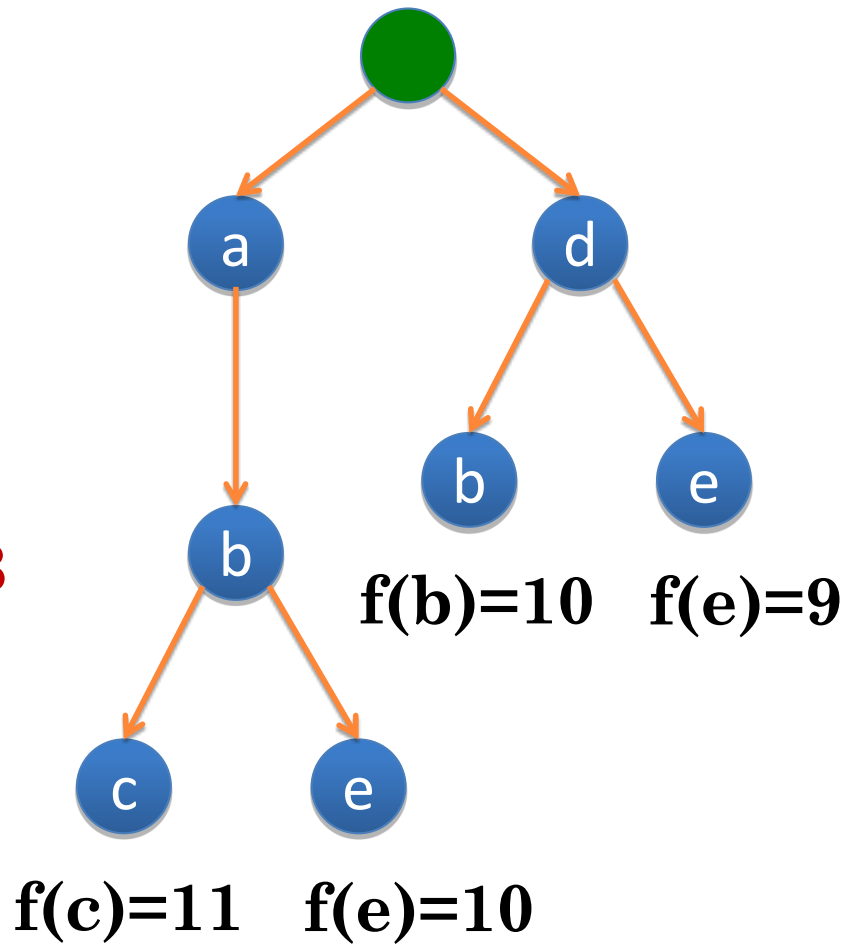
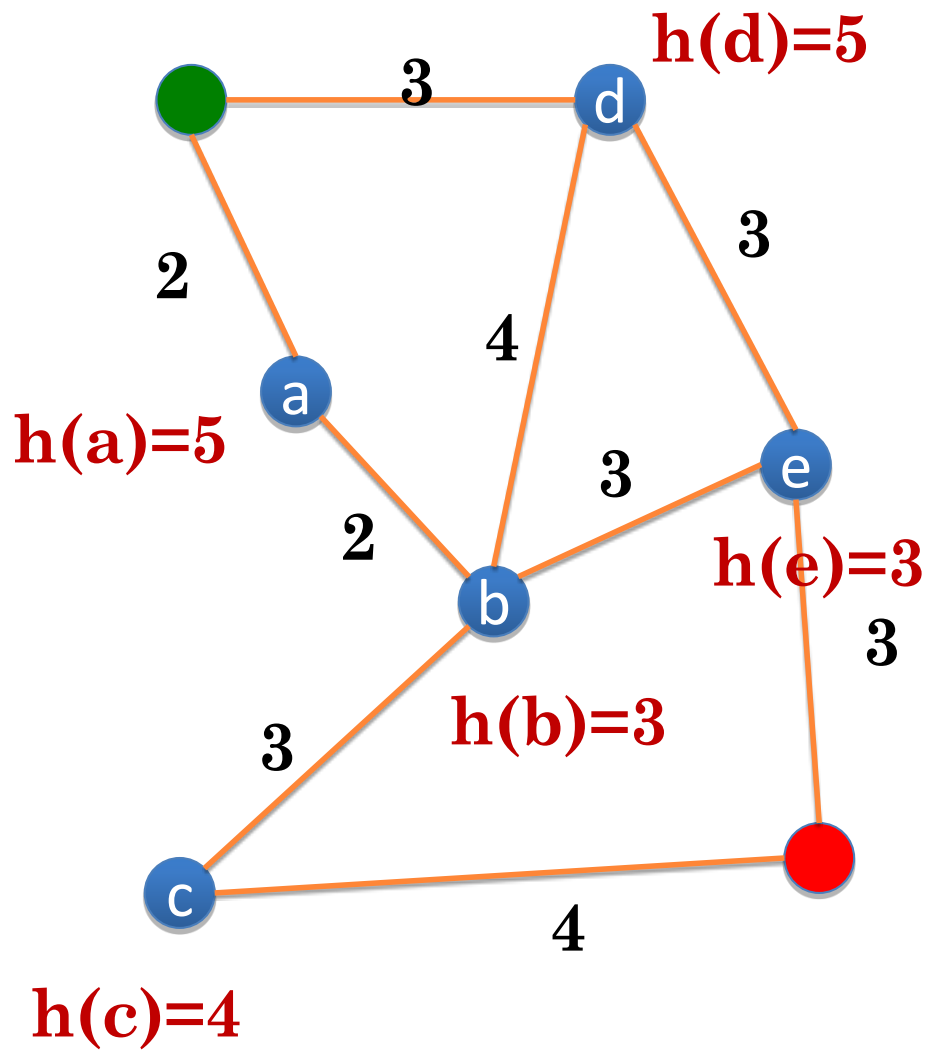


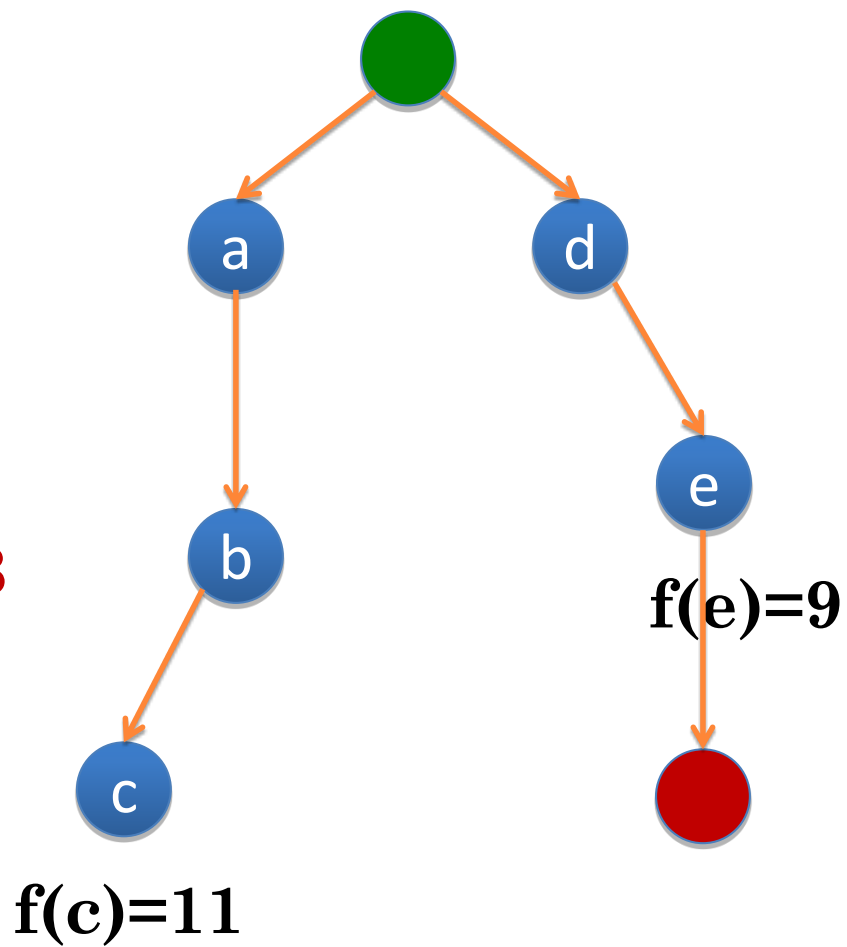
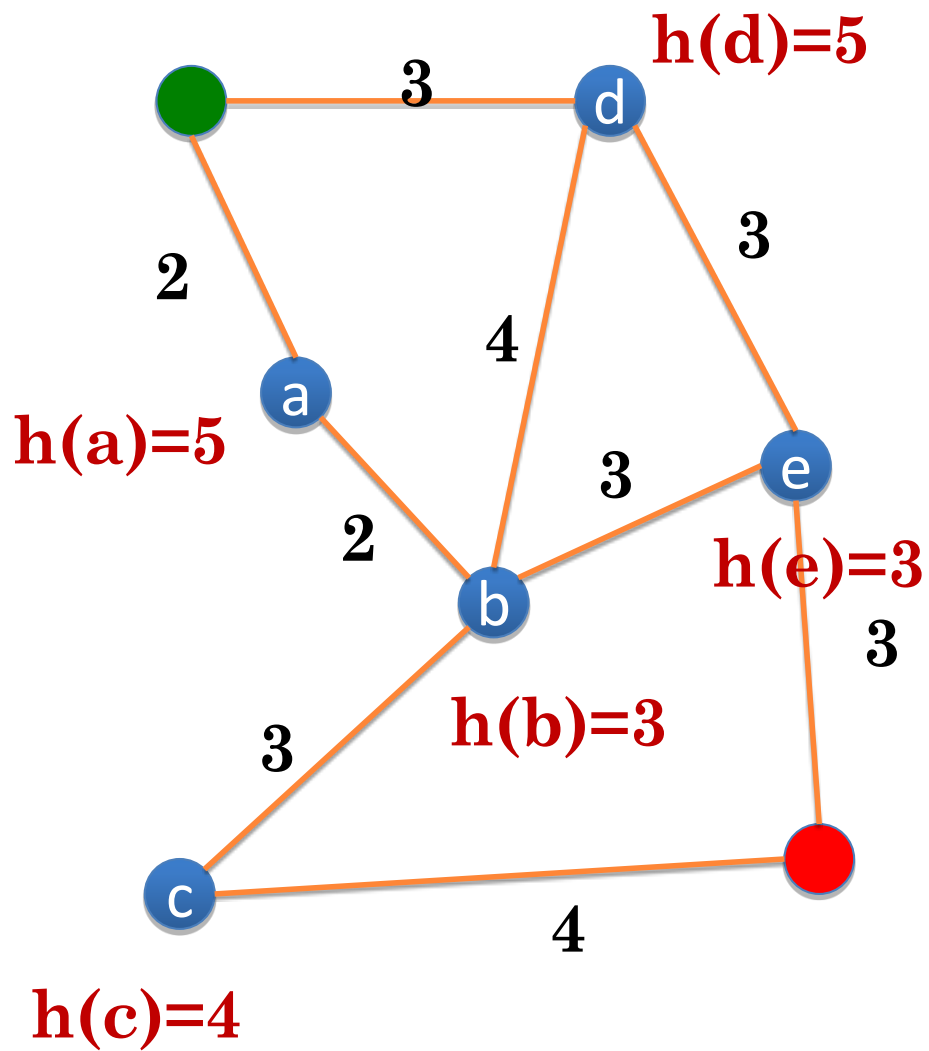


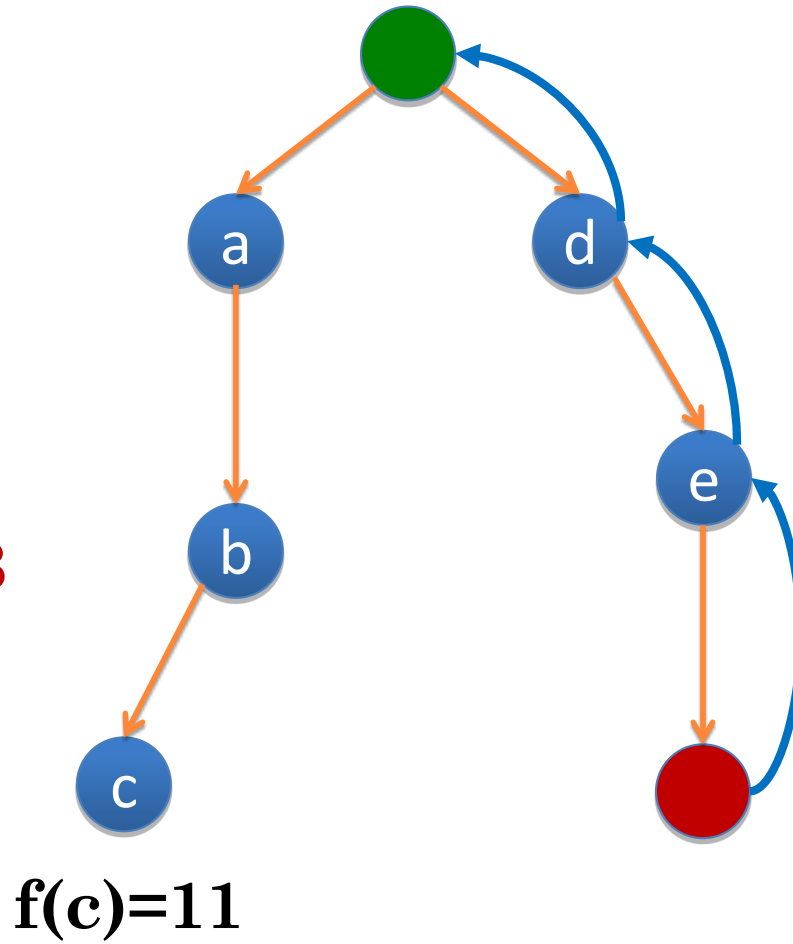
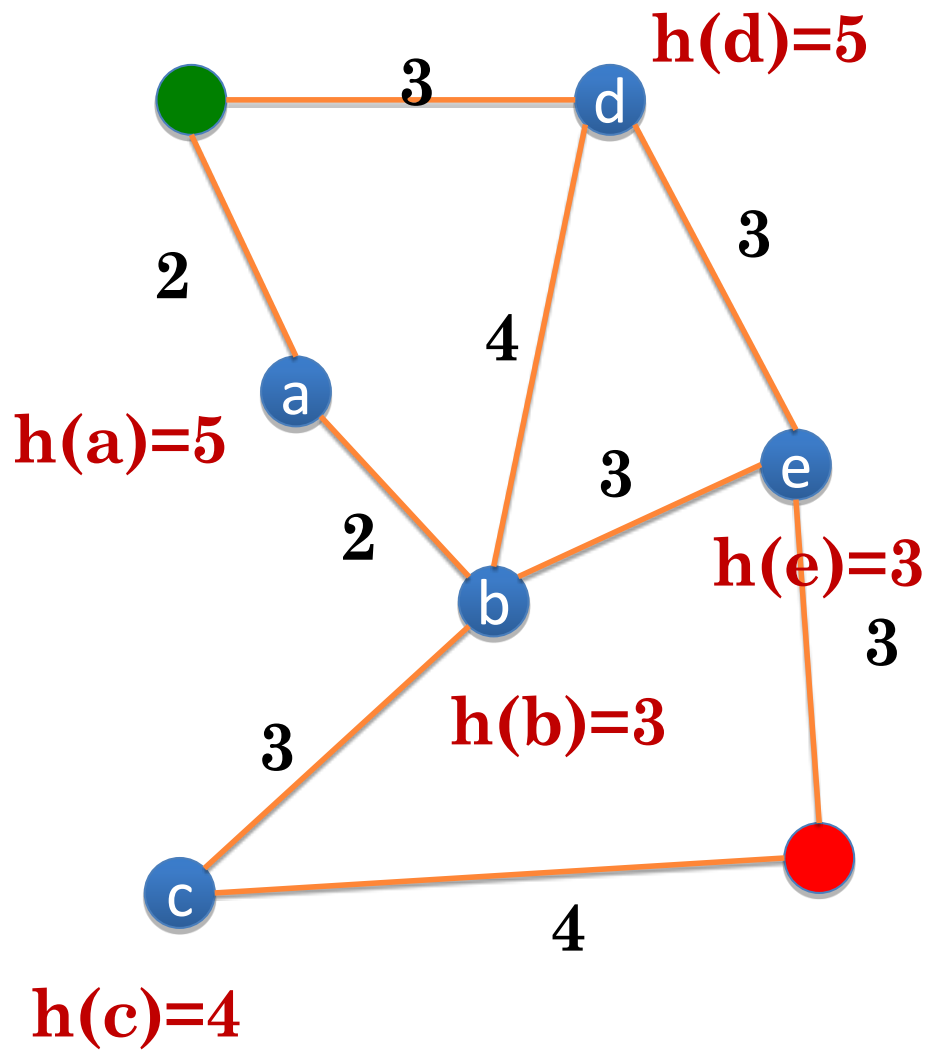


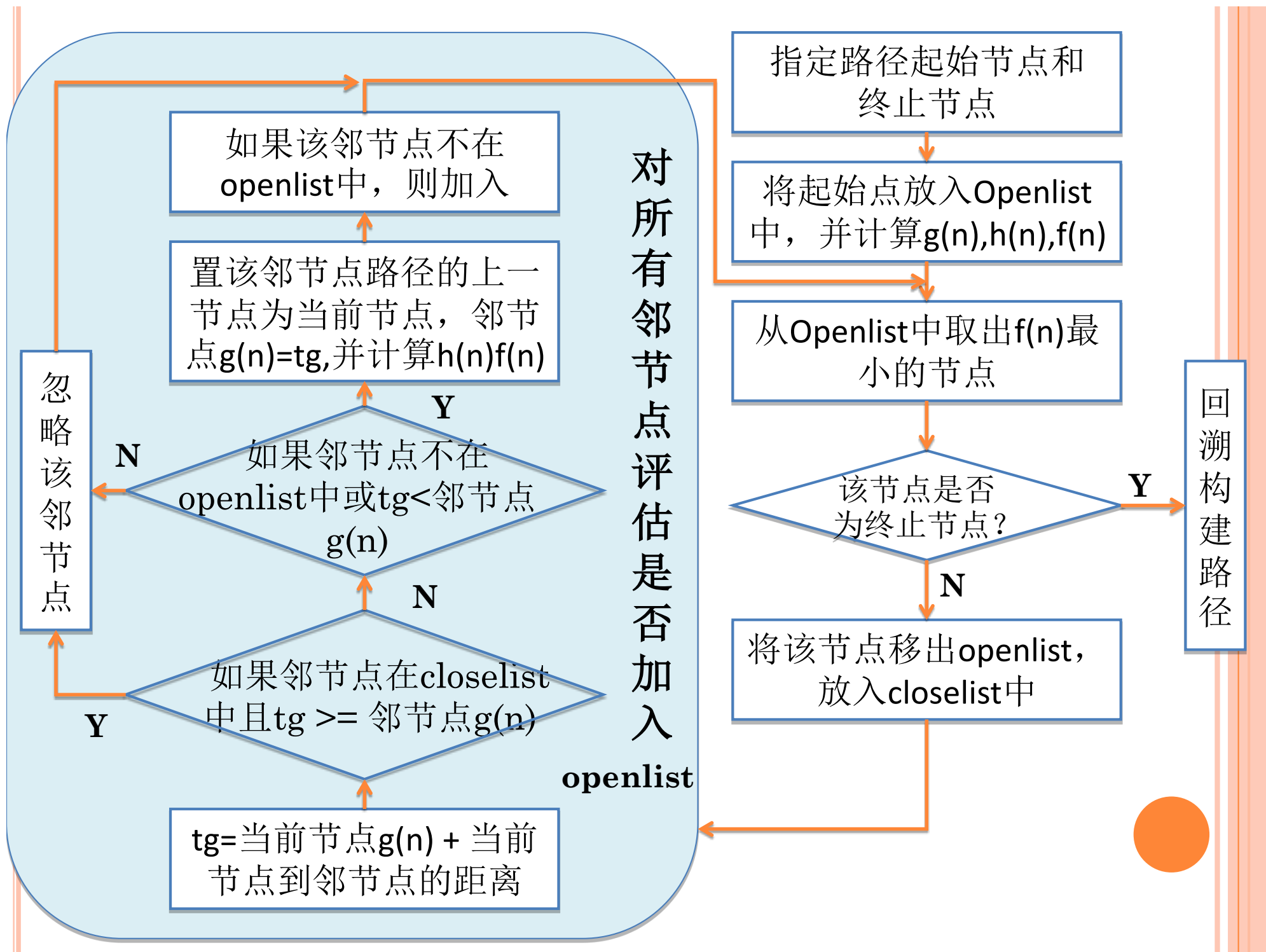


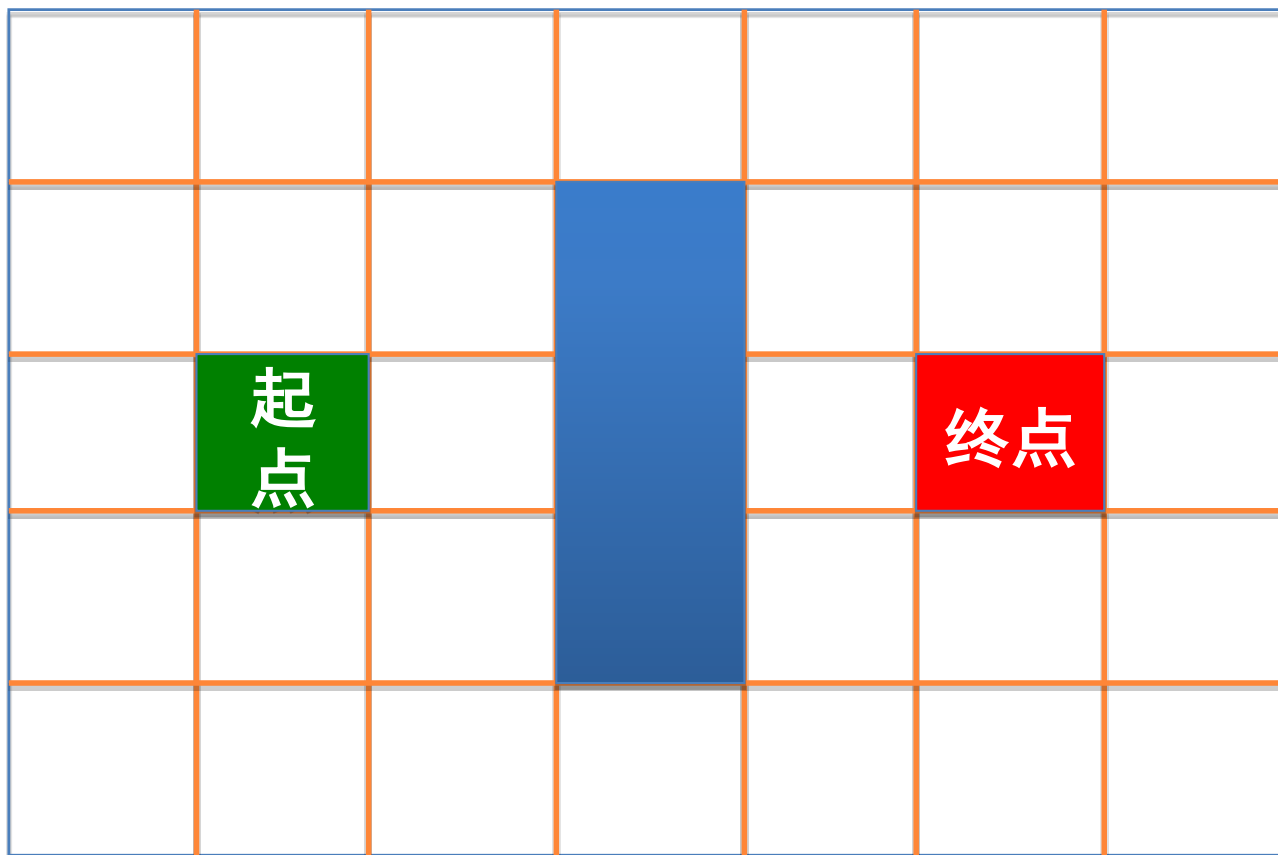












74	60	54				
14 60	10 50	14 40				
60	起点	40				
10 50		10 30				
74	60	54				
14 60	10 50	14 40				

终点





74	60	54				
14 60	10 50	14 40				
60	起点	40				
10 50		10 30				
74	60	54				
14 60	10 50	14 40				

终点



74 14 60	60 10 50	54 14 40	障碍区			
60 10 50	起点	40 10 30			终点	
74 14 60	60 10 50	54 14 40				

该节点被放入closelist，各邻节点已在openlist中，  
 $tg = 10 + 10$  或  $14$ ，大于各邻节点 $g(n)$ ，无操作



74 14 60	60 10 50	54 14 40				
60 10 50	起点	40 10 30				
74 14 60	60 10 50	54 14 40				



74 14 60	60 10 50	54 14 40	障碍区			
60 10 50	起点	40 10 30				
74 14 60	60 10 50	54 14 40				



74 14 60	60 10 50	54 14 40	障碍区			
60 10 50	起点	40 10 30				
74 14 60	60 10 50	54 14 40				
	88 28 60	74 24 50	68 28 40			
					终点	



	88 28 60	74 24 50	68 28 40			
74 14 60	60 10 50	54 14 40				
60 10 50	起点	40 10 30			终点	
74 14 60	60 10 50	54 14 40				
	78 28 60	74 24 50	68 28 40			



	80 20 60	74 24 50	68 28 40			
74 14 60	60 10 50	54 14 40				
60 10 50	起点	40 10 30			终点	
74 14 60	60 10 50	54 14 40				
94 24 70	80 20 60	74 24 50	68 28 40			



94 24 70	80 20 60	74 24 50	68 28 40			
74 14 60	60 10 50	54 14 40				
60 10 50	起点	40 10 30			终点	
74 14 60	60 10 50	54 14 40				
94 24 70	80 20 60	74 24 50	68 28 40			





94 24 70	80 20 60	74 24 50	68 28 40			
74 14 60	60 10 50	54 14 40				
60 10 50	起点	40 10 30			终点	
74 14 60	60 10 50	54 14 40				
94 24 70	80 20 60	74 24 50	68 28 40			



94 24 70	80 20 60	74 24 50	68 28 40			
74 14 60	60 10 50	54 14 40				
60 10 50	起点	40 10 30			终点	
74 14 60	60 10 50	54 14 40		62 42 20		
94 24 70	80 20 60	74 24 50	68 28 40	68 38 30		



94	80	74	68			
24 70	20 60	24 50	28 40			
74	60	54				
14 60	10 50	14 40				
60	起点	40		62	56	
10 50		10 30		52 10	56 0	
74	60	54		62	62	
14 60	10 50	14 40		42 20	52 10	
94	80	74	68	68		
24 70	20 60	24 50	28 40	38 30		



94 24 70	80 20 60	74 24 50	68 28 40			
74 14 60	60 10 50	54 14 40				
60 10 50	起 点	40 10 30				
74 14 60	60 10 50	54 14 40				
94 24 70	80 20 60	74 24 50	68 28 40	68 38 30		



# 路径成本的计算

- 欧式距离 (2-norm距离)
- 曼哈顿距离 (Manhattan distance, Block distance, 1-norm距离, L1 distance)
- 切比雪夫距离 (Chebychev distance, infinity norm距离)
- 明可夫斯基距离 (Minkowski距离)



# 欧式距离

## ○ 2-norm距离

$$\|\mathbf{x} - \mathbf{y}\|_2 = \sqrt{\sum_i (x_i - y_i)^2}$$

$$\mathbf{x} = (x_1, x_2, \dots, x_n)$$

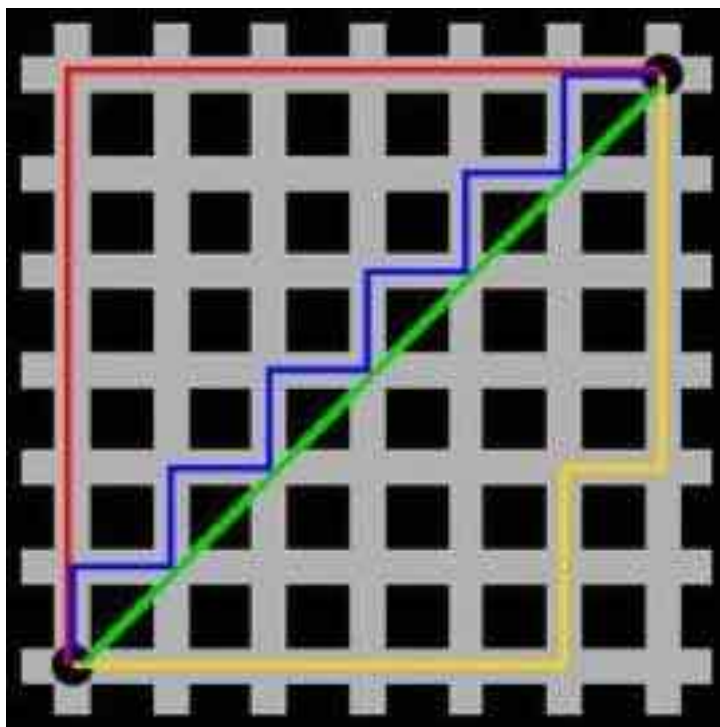
$$\mathbf{y} = (y_1, y_2, \dots, y_n)$$



# 曼哈顿距离

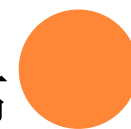
## ○ Manhattan distance

$$\|\mathbf{x} - \mathbf{y}\|_1 = \sum_i |x_i - y_i|$$



用以标明两个点上在标准  
坐标系上的绝对轴距总和

绿线为欧氏距离  
红线为曼哈顿距离  
黄蓝线为等价曼哈顿距离




# 切比雪夫距离

- Chebychev distance

$$\|\mathbf{x} - \mathbf{y}\|_{\infty} = \max_i |x_i - y_i|$$



	a	b	c	d	e	f	g	h	
8	5	4	3	2	2	2	2	2	8
7	5	4	3	2	1	1	1	2	7
6	5	4	3	2	1		1	2	6
5	5	4	3	2	1	1	1	2	5
4	5	4	3	2	2	2	2	2	4
3	5	4	3	3	3	3	3	3	3
2	5	4	4	4	4	4	4	4	2
1	5	5	5	5	5	5	5	5	1
	a	b	c	d	e	f	g	h	

起源于国际国王行走距离计算  
可用于仓储物流距离计算



# 明可夫斯基(Minkowski)距离

- 两个个体各个变量值绝对差的k次幂总和的k次方根

$$Minkowski(\mathbf{x}, \mathbf{y}) = \sqrt[k]{\sum_i |x_i - y_i|^k}$$

k可以任意指定

$k = 1$ , 曼哈顿距离

$k = 2$ , 欧氏距离

$k \rightarrow \infty$ , 切比雪夫距离

