华东师范大学计算机科学与技术学院
School of Computer Science and Technology

2024年春季学期

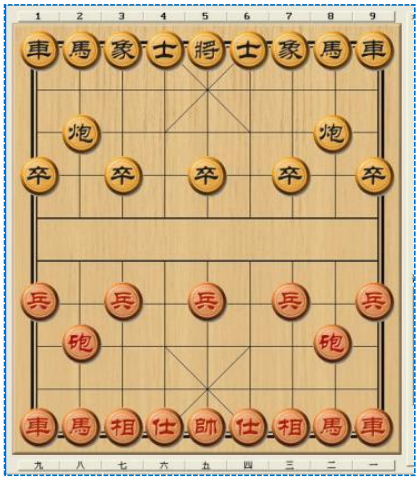
《人工智能》

第2讲：一般图搜索

搜索方法

- 什么是搜索？


Сталинград
Солодча
Серафимович
Мар.Ивановка
Трехостровская
Суровикино
Гавриловка
Иванцово
Логовский
Цаца
Васильевка
Абганерово
Райгород
Каменный Яр
16ВА
21А
1Гв.А
4ТА
24А
66А
62А
1ТА
6А
64А
57А
8ВА




华东师范大学(中北校区)
华东师范大学(闵行校区)
上海市
909路/224路 → 上嘉线/上青线 → 上奉专线
地铁3号线 → 上奉专线
地铁3号线 → 958路
地铁4号线 → 地铁1号线 → 地铁5号线 → 江川3路

搜索方法

- 什么是搜索？
  - 搜索算法是利用计算机的高性能来有目的的穷举一个问题解空间的部分或者所有的可能情况，从而求出问题的解的一种计算机算法。

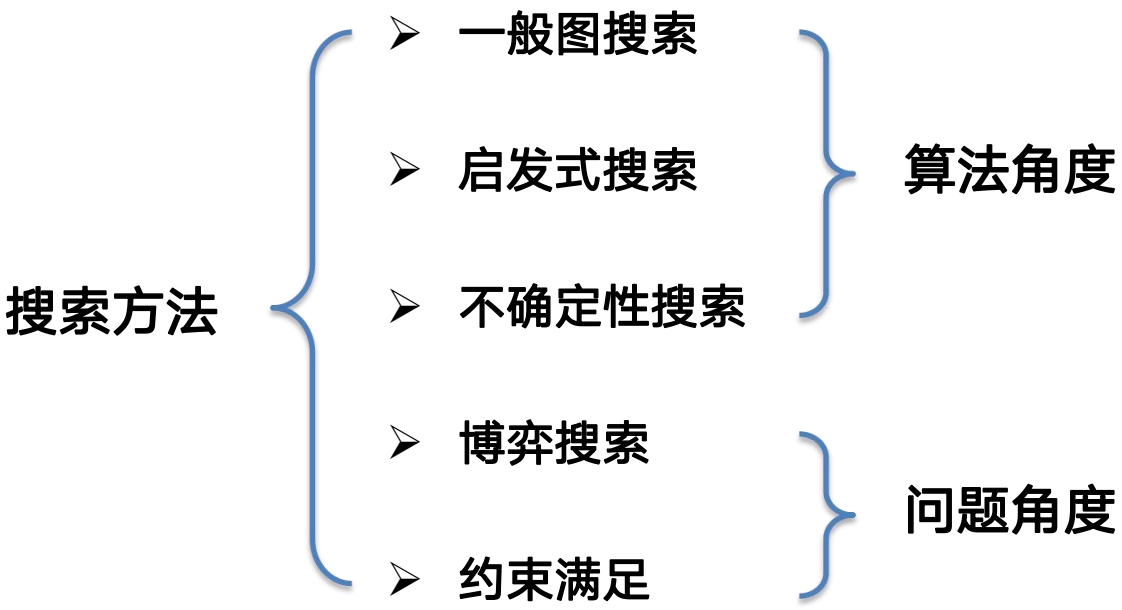
计算机算得快
模仿人类智能

具有求解目标
有的放矢

针对某个具体问题
非泛指

第2章 一般图搜索

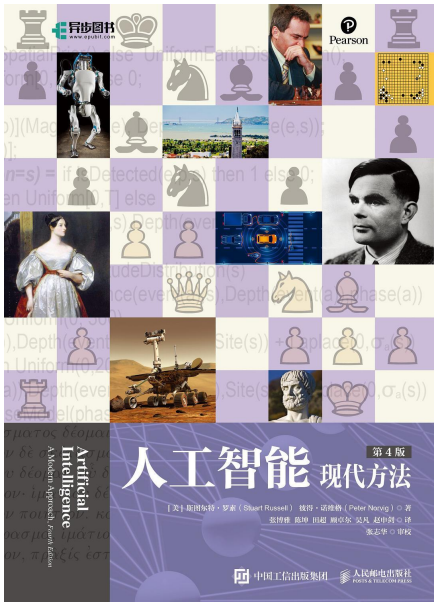
- 有哪些搜索方法？



第2章 一般图搜索

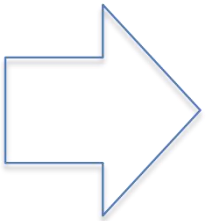
- **主要内容**
  - 2.1 图搜索基本概念
  - 2.2 一般图搜索算法
  - 2.3 九宫格游戏求解
  - 2.4 本章小结
- **参考书目**
  - 《人工智能：现代方法（第4版）》（美）罗素，（美）诺维格，人民邮电出版社，2022。

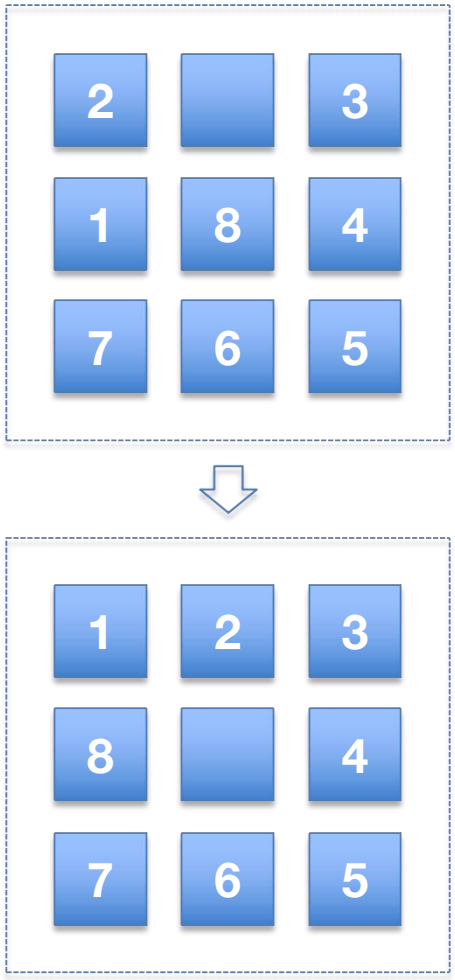
    Ch3: pp.54-73.



2.1 图搜索基本概念

- 九宫格游戏



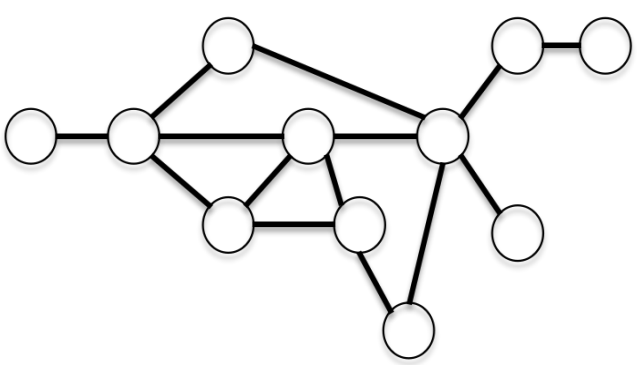


人如何求解？

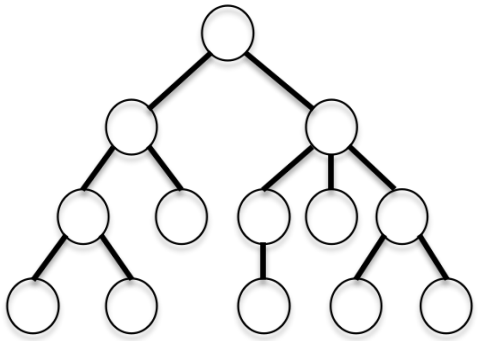
计算机如何求解？

2.1 图搜索基本概念

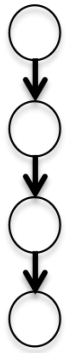
- 图搜索是搜索过程可以用图结构的形式呈现的一类搜索算法。图可以更加形象与清晰地描述搜索过程。
- 在计算机科学中，一个图就是一些顶点的集合，这些顶点通过一系列边连接。顶点用圆圈表示，边就是这些圆圈之间的连线，顶点之间通过边连接。



Graph



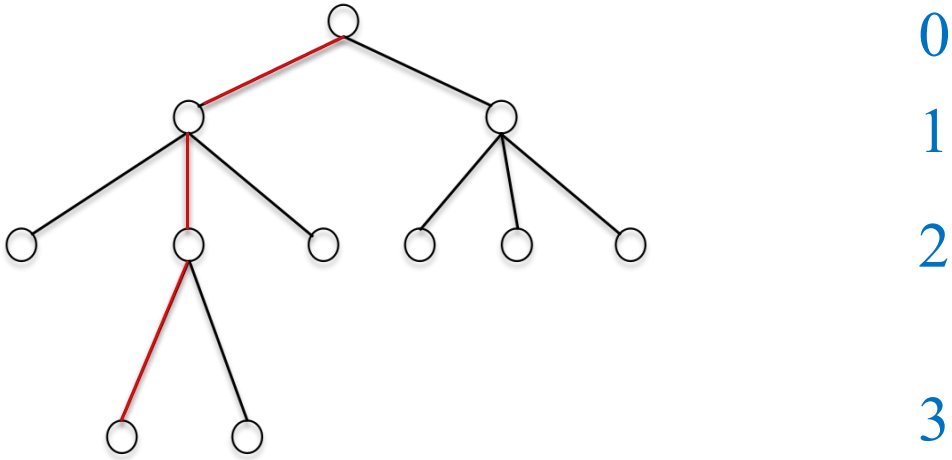
Tree



Linked List

2.1 图搜索基本概念

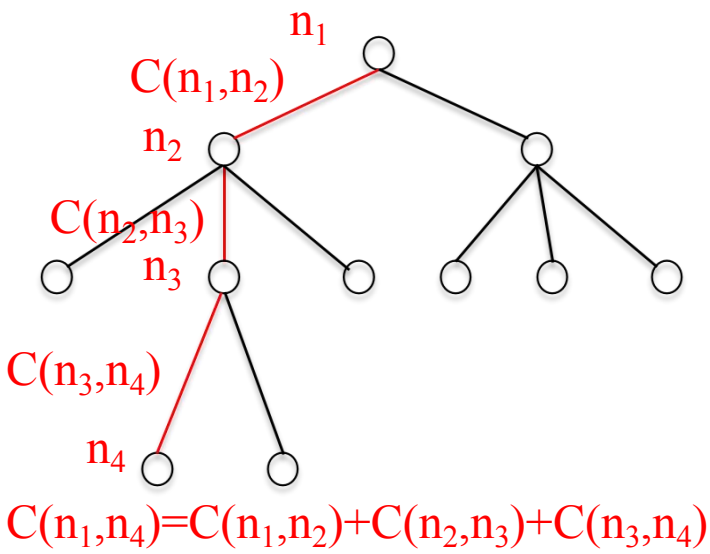
- **节点深度：**
  - 根节点深度=0
  - 其它节点深度=父节点深度+1



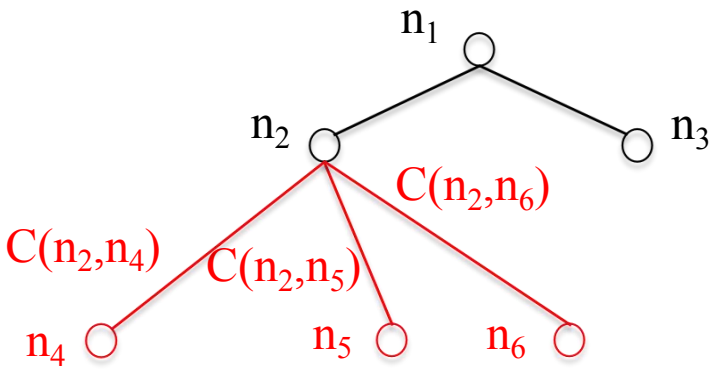
- **路径**
  - 设一节点序列为(n_0, n_1,…,n_k)，对于i=1,…,k，若节点n_{i-1}具有一个后继节点n_i，则该序列称为从n_0到n_k的路径。

2.1 图搜索基本概念

- **路径的耗散值**
  - 一条路径的耗散值等于连接这条路径各节点间所有耗散值的总和。用$C(n_i, n_j)$表示从n_i到n_j的路径的耗散值。

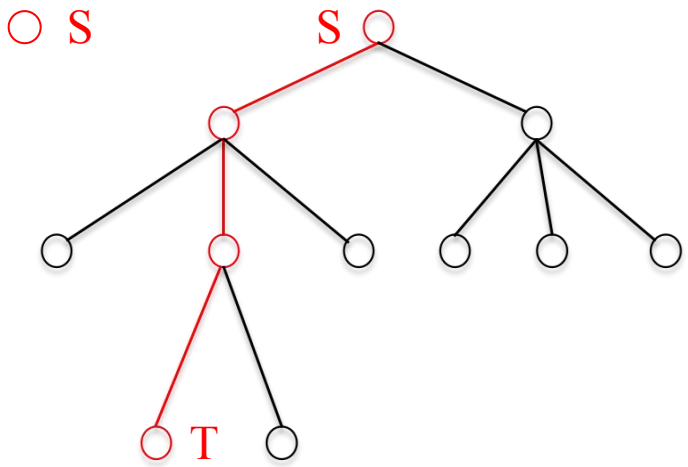


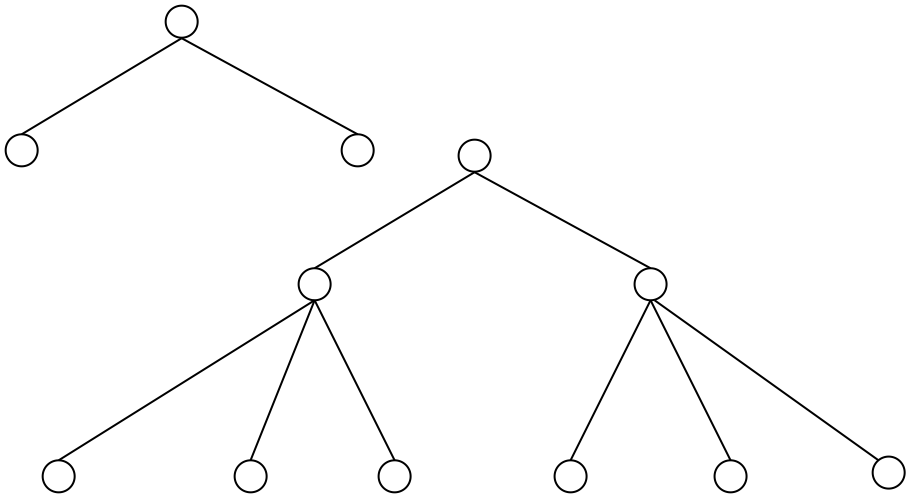
- **扩展一个节点**
  - 生成出该节点的所有后继节点，并给出它们之间的耗散值。这一过程称为“扩展一个节点”。



2.1 图搜索基本概念

- **初始状态**
  - 问题求解之前的状态，一般是一个空状态，用符号S表示，也用作表示初始节点。
- **目标状态**
  - 包含求解目标的一种状态，用符号T表示，也用作表示一个目标节点。（T到S的反推代表一个解）
- **问题状态**
  - 问题求解过程中的一种状态，某种中间过程的一个呈现，其对应于某个图G。
- **状态空间**
  - 问题求解过程中所有可能状态的集合





2.2 一般图搜索算法

- **基本思想（概念算法，从计算机角度思考）**
  - 从给定的**初始状态**出发
  - 一步一步选择节点**扩展**，得到新的状态
  - 重复上述步骤，直到扩展**目标状态**（T到S的路径即为所求解），或者无节点可扩展（该问题无解）
- **三个数据结构**
  - **Open表**：　存储还未扩展的节点
  - **Closed表**：　存储已经扩展过的节点
  - **状态图G**：　当前已经扩展出来节点构成的当前状态

2.2 一般图搜索算法

- 算法框架（伪代码表示）

1 $G=G_0$ (G_0=s), Open:=(s), Closed:=( ); // 初始化

2 Loop: If Open=( ) Then Exit(Fail); // 失败停止

3 n:=First(Open), Remove(n, Open), Add(n, Closed); // 更新数据结构

4 If Goal(n)，Then Exit(Success); // 成功停止

5 Expand(n)→{m_i}, 计算耗散值, G:=Add(m_i, G); // 扩展节点，更新状态

6 标记和修改指针： // 修改指针（最优解）

Add(m_j, Open), 并标记m_j到n的指针； // m_j 表示新扩展出来的节点

计算是否要修改m_k、m_l到n的指针； // m_k在Open表中， m_l在Closed表中

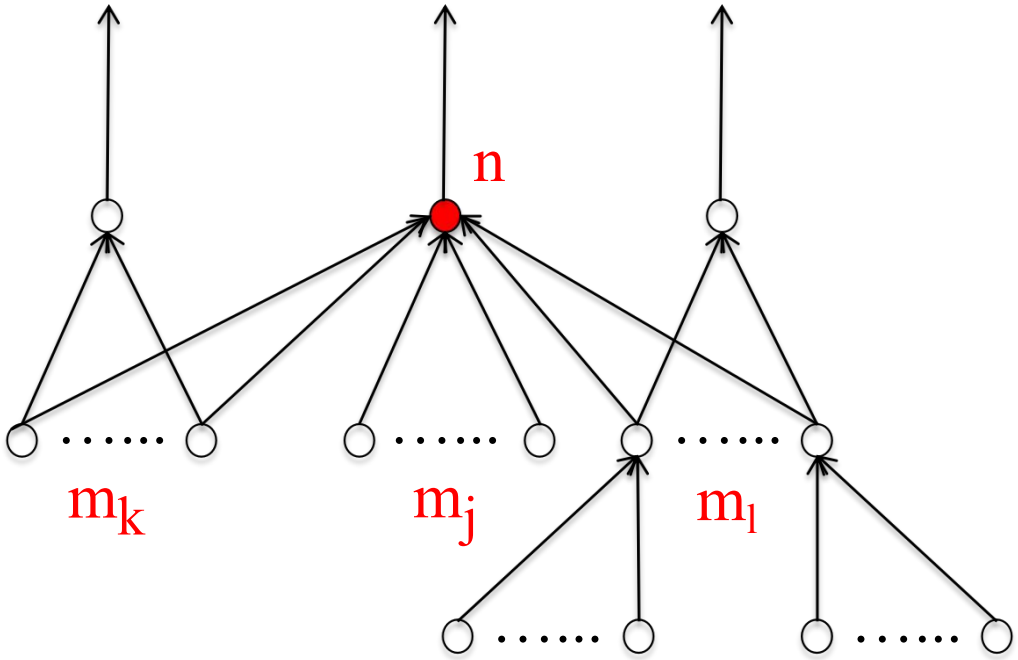
计算是否要修改m_l的子孙节点的指针；

7 对Open中的节点按某种原则重新排序； // 保证找到最优解

8 Go Loop；

2.2 一般图搜索算法

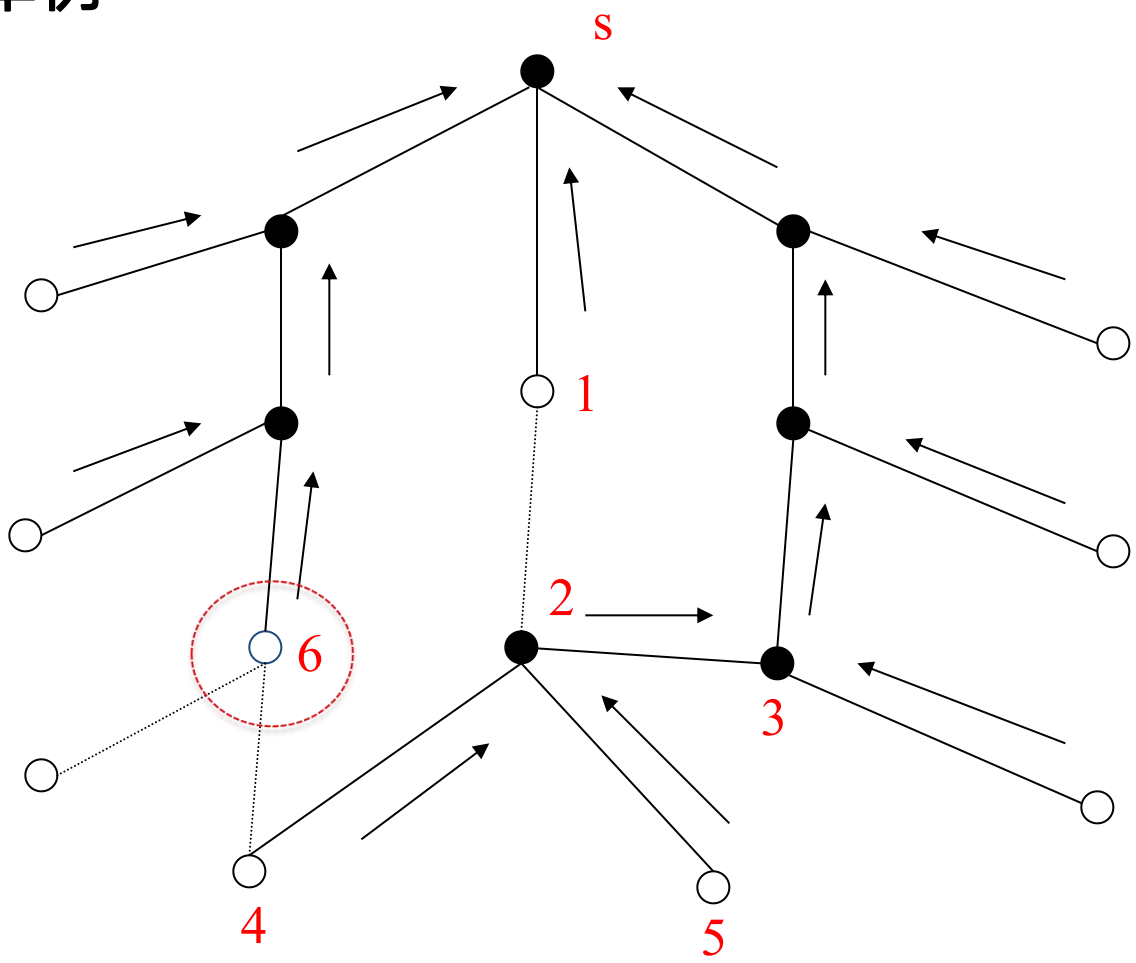
- 节点类型



- m_k在Open表中
- m_j新扩展出来（Open和Close表中未出现过）
- m_l在Closed表中

2.2 一般图搜索算法

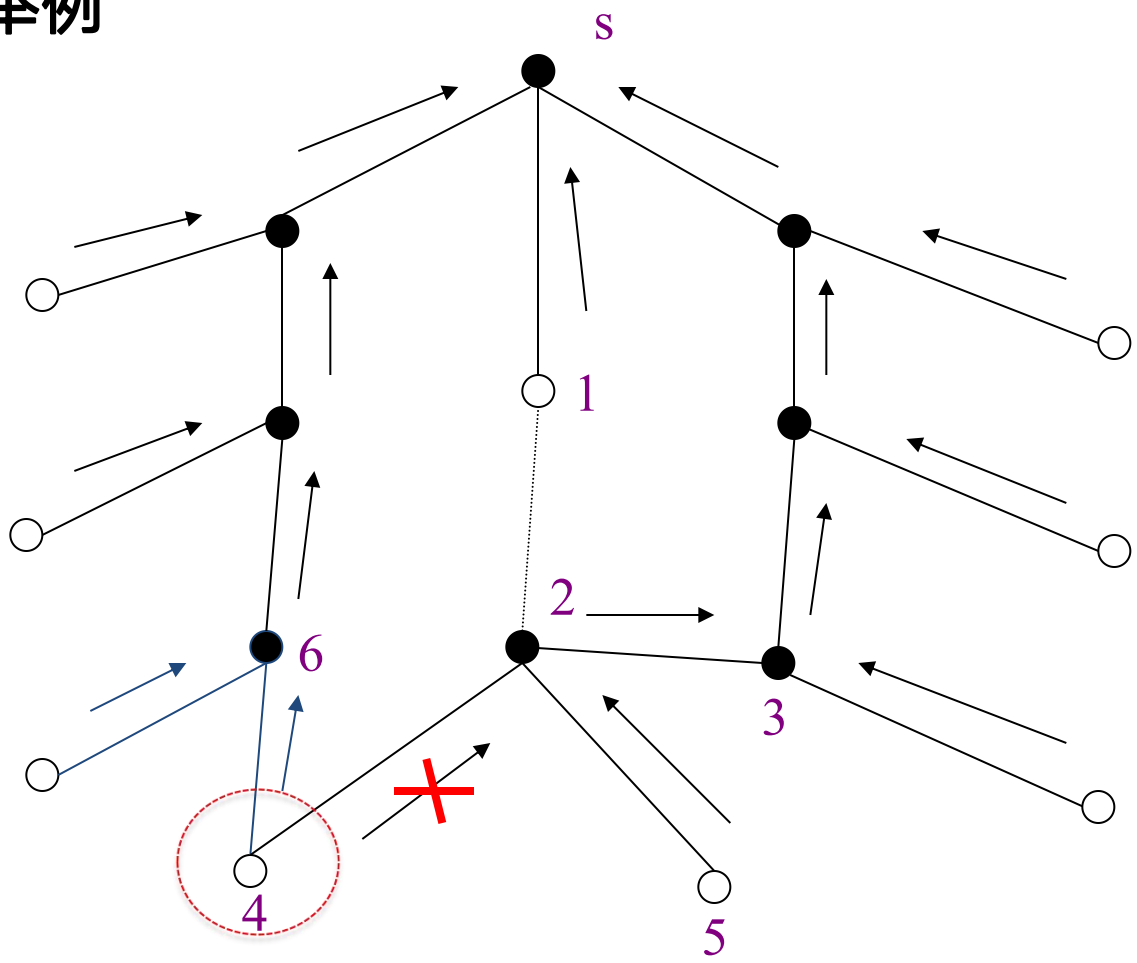
- 修改指针举例



- 相邻节点之间耗散值：1
- 下一个扩展节点：6

2.2 一般图搜索算法

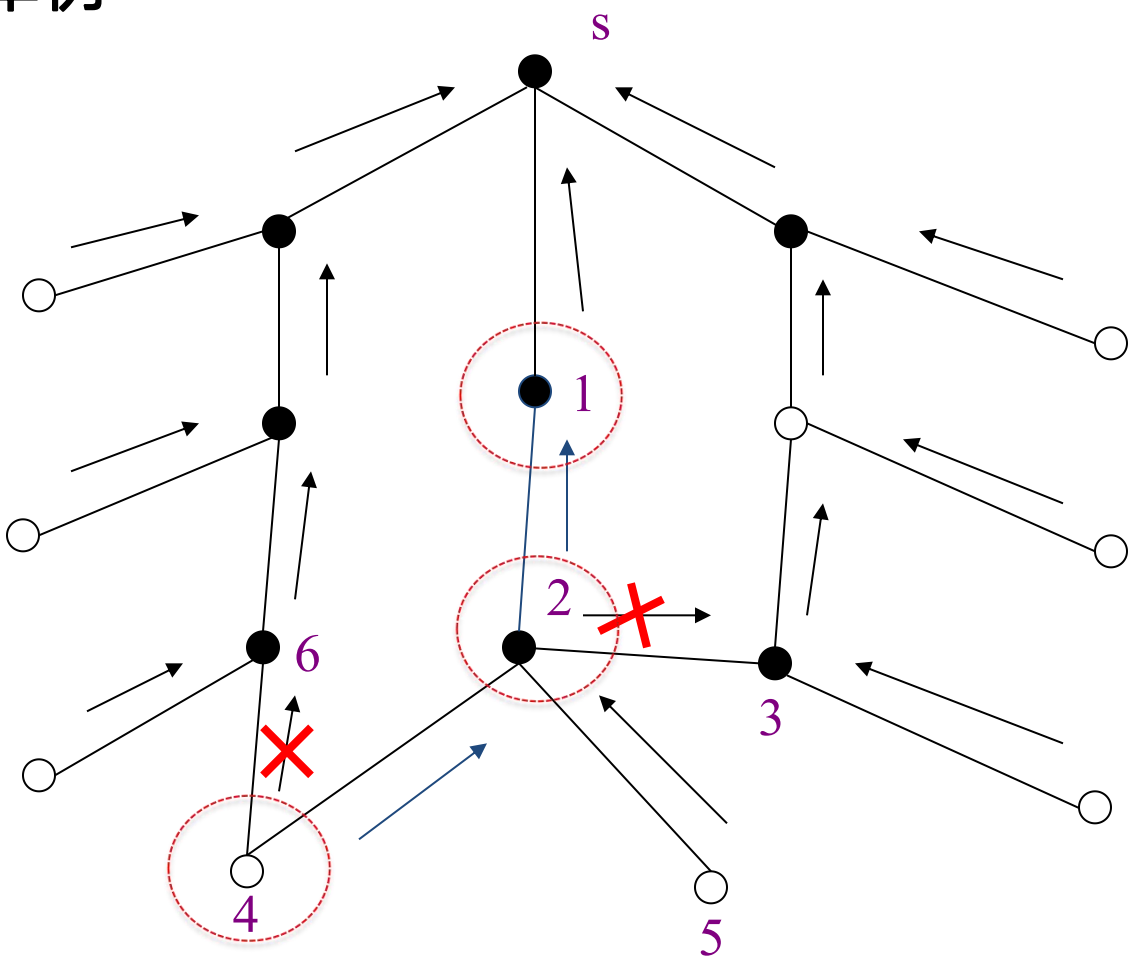
- 修改指针举例



➢ 节点4在Open表中，被6再次扩展，有可能找到一条新路径

2.2 一般图搜索算法

- 修改指针举例

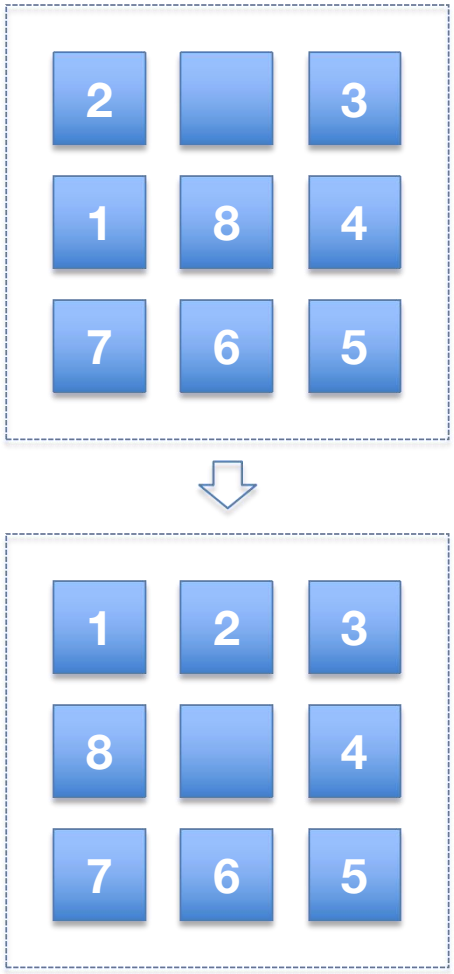


➢ 扩展节点1，节点2在Closed表中，找到新路径

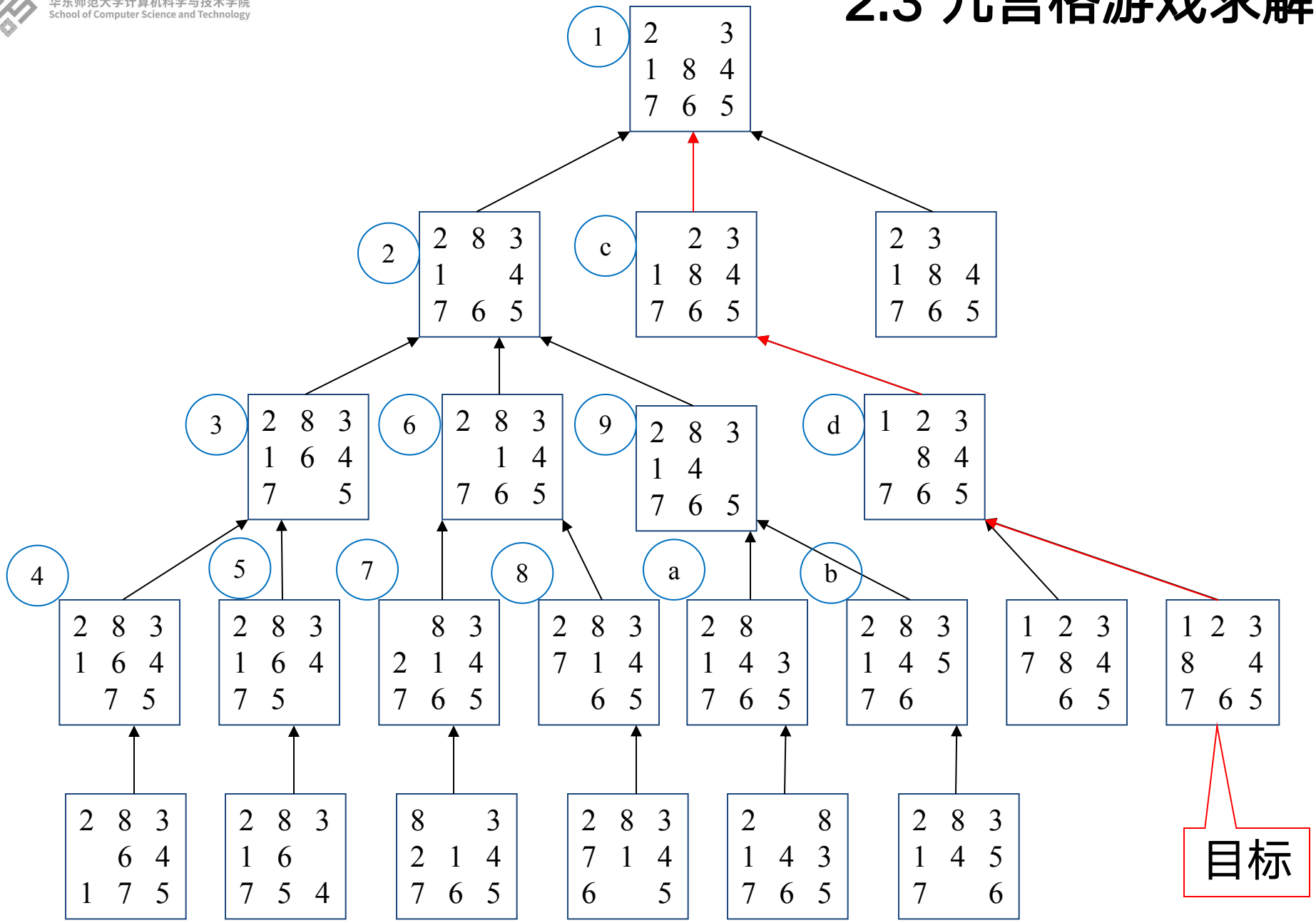
➢ 节点2的后继节点4在Open表中，找到更优路径

2.3 九宫格游戏求解

- 初始状态
- 目标状态
- 扩展规则
  - 空格挪动顺序：上->下->左->右
- 耗散值
  - 相邻节点之间耗散值1
- 目标
  - 挪动步骤最少



2.3 九宫格游戏求解


1
2 3
1 8 4
7 6 5
2
2 8 3
1 4
7 6 5
c
2 3
1 8 4
7 6 5
2 3
1 8 4
7 6 5
3
2 8 3
1 6 4
7 5
6
2 8 3
1 4
7 6 5
9
2 8 3
1 4
7 6 5
d
1 2 3
8 4
7 6 5
4
2 8 3
1 6 4
7 5
5
2 8 3
1 6 4
7 5
7
8 3
2 1 4
7 6 5
8
2 8 3
7 1 4
6 5
a
2 8
1 4 3
7 6 5
b
2 8 3
1 4 5
7 6
1 2 3
7 8 4
6 5
1 2 3
8 4
7 6 5
2 8 3
6 4
1 7 5
2 8 3
1 6
7 5 4
8 3
2 1 4
7 6 5
2 8 3
7 1 4
6 5
2 8
1 4 3
7 6 5
2 8 3
1 4 5
7 6
目标

2.3 九宫格游戏求解

- **问题**

1. 上述算法中Open表的排序规则是什么？

2. 深度优先搜索算法DFS和宽度优先搜索算法BFS可以简单描述如下，如何采用一般图搜索框架描述DFS和BFS？

```
Void DFS(Node m){
   For(m的所有后继节点n){
      If(n没有被访问){
         If(n是目标节点)
            Return true;
         标记n;
         DFS(n);
      }
   }
}
```

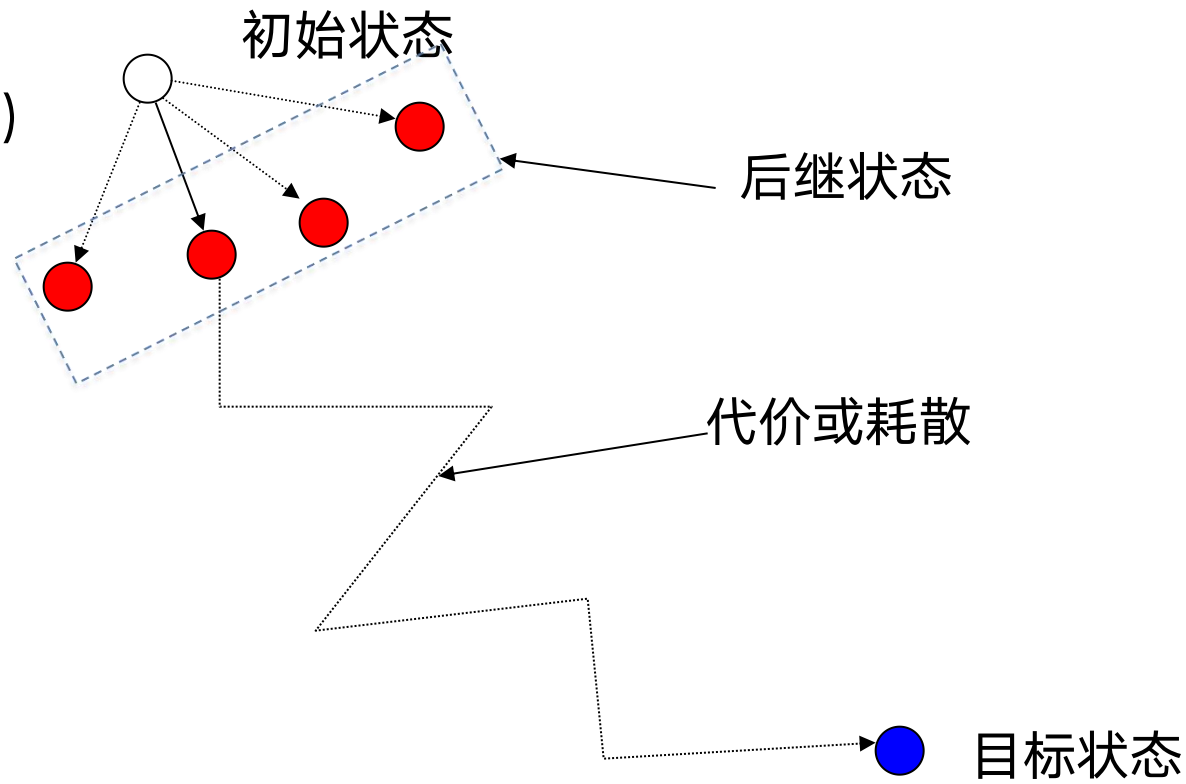
DFS

```
初始化队列Q={S},标记S已访问;
While(Q非空){
   取出Q队首元素n，n出Q队列;
   If(n是目标节点)
      Return true;
   所有n的后继节点进入Q队列尾部;
   标记n的后继节点已访问;
}
```

BFS

2.4 本章小结

- **一般图搜索的基本要素**
  - 初始状态
  - 目标状态
  - 扩展规则（后继函数）
  - 耗散函数及排序规则



2.4 本章小结

- **一般图搜索是一类无信息搜索**
  - 与所求解问题无关
  - 深度优先搜索
  - 宽度优先搜索
- **一般图搜索算法是很多算法的基础**
  - 通常通过在搜索前，根据条件降低搜索规模
  - 根据问题的约束条件进行剪枝
  - 利用搜索过程中的中间解，避免重复计算
  - 利用问题启发式信息，减少搜索空间

后续章节
陆续展开

谢谢