



问题解答

Dijkstra负权重问题

最短路径中的优先队列

【大作业】二〇四八

数据结构与算法 (Python) -18/0512

陈斌 gischen@pku.edu.cn 北京大学地球与空间科学学院

目录

- › 问题解答
- › 【大作业】二〇四八
- › 再也没有KHC了！



问题解答

- › 有不少使用到的接口(color, distance等)在原始vertex类中都没有实现,能详细说明一下吗?

.....就是增加了2个实例属性, get/set, 没有特殊处理

```
61     def setColor(self,color):
62         self.color = color
63
64     def setDistance(self,d):
65         self.dist = d
66
67     def setPred(self,p):
68         self.pred = p
```

```
45 class Vertex:
46     def __init__(self,num):
47         self.id = num
48         self.connectedTo = {}
49         self.color = 'white'
50         self.dist = sys.maxsize
51         self.pred = None
52         self.disc = 0
53         self.fin = 0
```

```
82     def getPred(self):
83         return self.pred
84
85     def getDistance(self):
86         return self.dist
87
88     def getColor(self):
89         return self.color
```

问题解答

- › **课件703中iter方法迭代的为何是Vertex类,而不是key值?**
 -我们希望对图Graph迭代的时候是一个个的节点Vertex, 而非仅仅key
 -图 $G=\langle V, E \rangle$, 所以希望迭代出Vertex
- › **课件703中nv=...add, 赋值什么用?**
 -好像是没什么用, 可以去掉nv=

问题解答

- › 单词关系图复杂度为何不是 $O(V)$? 每个词最多关联4个桶, 每个桶最多装26个点, 都是常数级别吧?

.....这里 $O(V+E)$ 是指通常的BFS而言, 并非针对词梯这个特定问题。

词梯问题是用BFS来解决的

广度优先搜索算法分析

❖ BFS算法主体是两个循环的嵌套

while循环对每个顶点访问一次, 所以是 $O(|V|)$

而嵌套在**while**中的**for**, 由于每条边只有在其起始顶点u出队的时候才会被检查一次

而每个顶点最多出队1次, 所以**边最多被检查1次**, 一共是 $O(|E|)$

综合起来BFS的时间复杂度为 $O(|V|+|E|)$

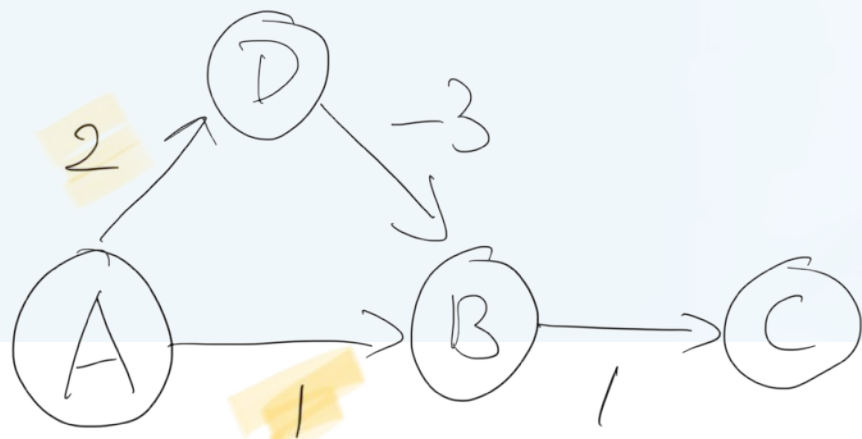
问题解答

- › 能说说最小路径算法如果出现负权重的情况为什么不一定能得到正确结果吗？

.....Dijkstra算法是一个贪心算法，优先队列会按照距离值，每次移走一个距离值最小的节点，而且不再更新距离加回队列

默认A-B如果是A连接所有节点的最小权重的话，那么B就会以这个权重作为最短路径，出队。

如果权重全都非负，这个贪心就成立，否则就不成立。



从 A-B 已有 A-B
A-D
最短为 1, B 已
标黑出队

问题解答

› 详细讲一下优先队列

.....优先队列里，每次出队的是最小的key，这就叫优先队列

› 最短路径问题中decreasekey是什么？

.....这个优先队列以amt作为优先级，携带val(节点的id)

decreasekey就是通过val找到队列里的数据项，把其中优先级减少到amt

```
73 def decreaseKey(self, val, amt):
74     # this is a little wierd, but we need to find the heap thi
75     # looking at its value
76     done = False
77     i = 1
78     myKey = 0
79     while not done and i <= self.currentSize:
80         if self.heapArray[i][1] == val:
81             done = True
82             myKey = i
83         else:
84             i = i + 1
85     if myKey > 0:
86         self.heapArray[myKey] = (amt, self.heapArray[myKey][1])
87         self.percUp(myKey)
```

找节点

减少，所以只需要上浮

问题解答

› 连通分支、连通分量是啥?网上查到的是关于数学的问题

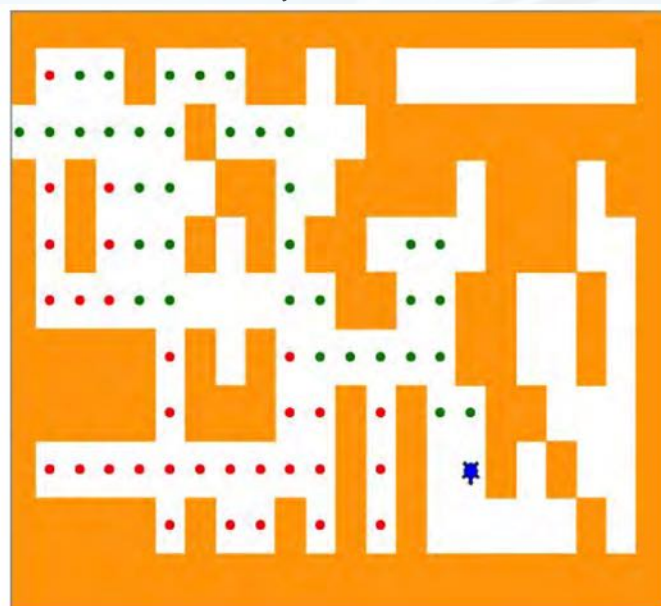
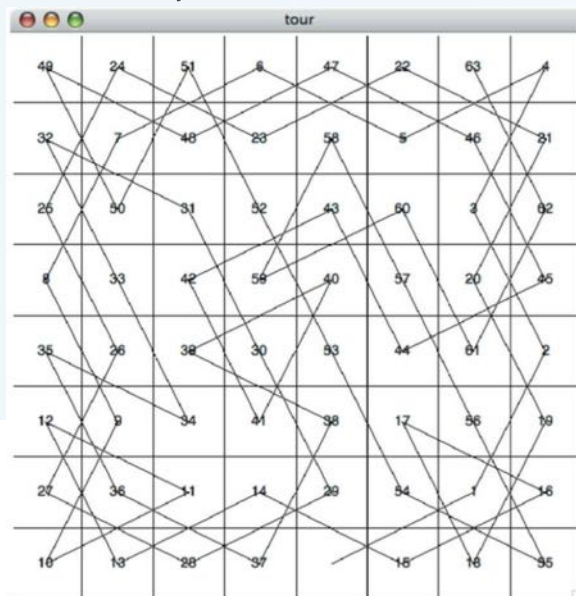
图 G 的子图 C , C 中的任意两个顶点 x, y 之间都可达, 称为连通的

而且如果再往 C 中加进任意顶点, C 就不连通了, 这样的 C 叫做连通分支

› 骑士周游问题和之前讲过的探索迷宫问题有何异同?

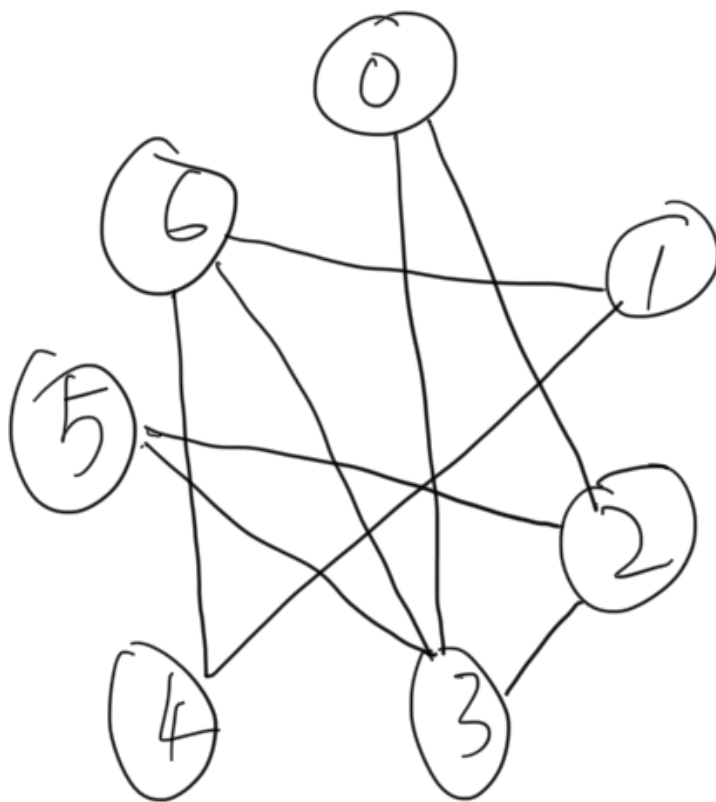
一样都可以用DFS来求解

把迷宫的每个格子对应节点, 相邻格子能到达的加一条边, 即可成为图



7 单选 (2分) 有一个无向图的邻接矩阵如下图所示。问此无向图有 () 条边, () 个连通分支。

	0	1	2	3	4	5	6
0	0	0	1	1	0	0	0
1	0	0	0	0	1	0	1
2	1	0	0	1	0	1	0
3	1	0	1	0	0	1	1
4	0	1	0	0	0	0	1
5	0	0	1	1	0	0	0
6	0	1	0	1	1	0	0



- ☒ A. 9, 1
- ☐ B. 9, 2
- ☐ C. 18, 1
- ☐ D. 18, 2

【H5】修订1次

› 鉴于许多同学【H5】作业出现的捶胸现象

少1个空格

少1个括号

少1个getRoot

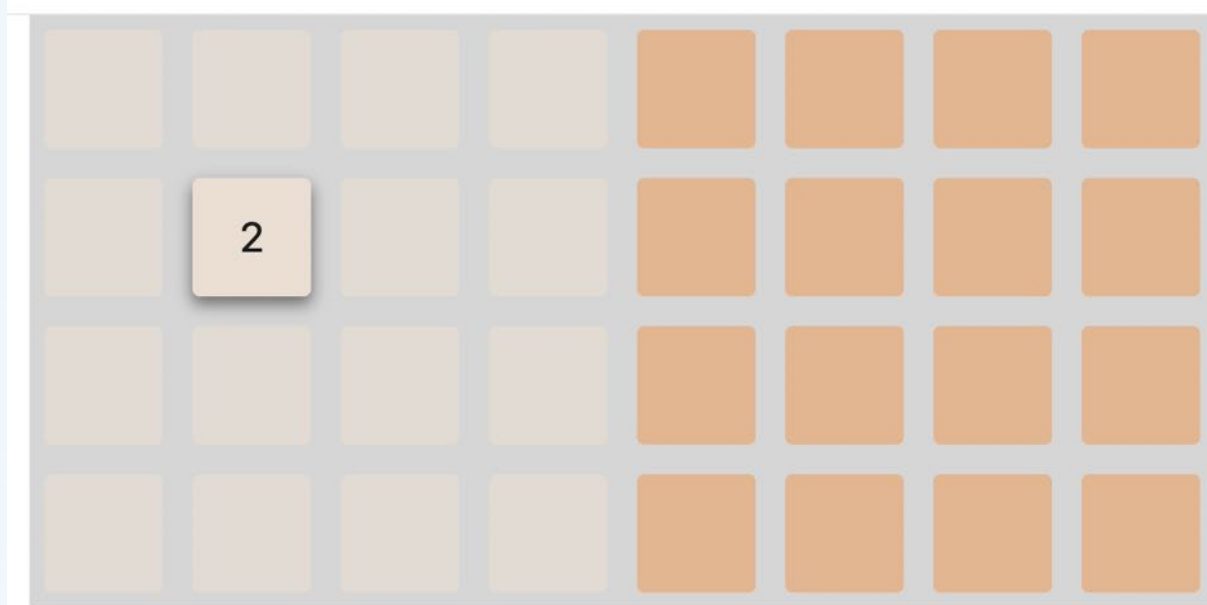
.....

› 【H5】限时重开24小时，到明晚6点，修订1次

这次修订不算补交

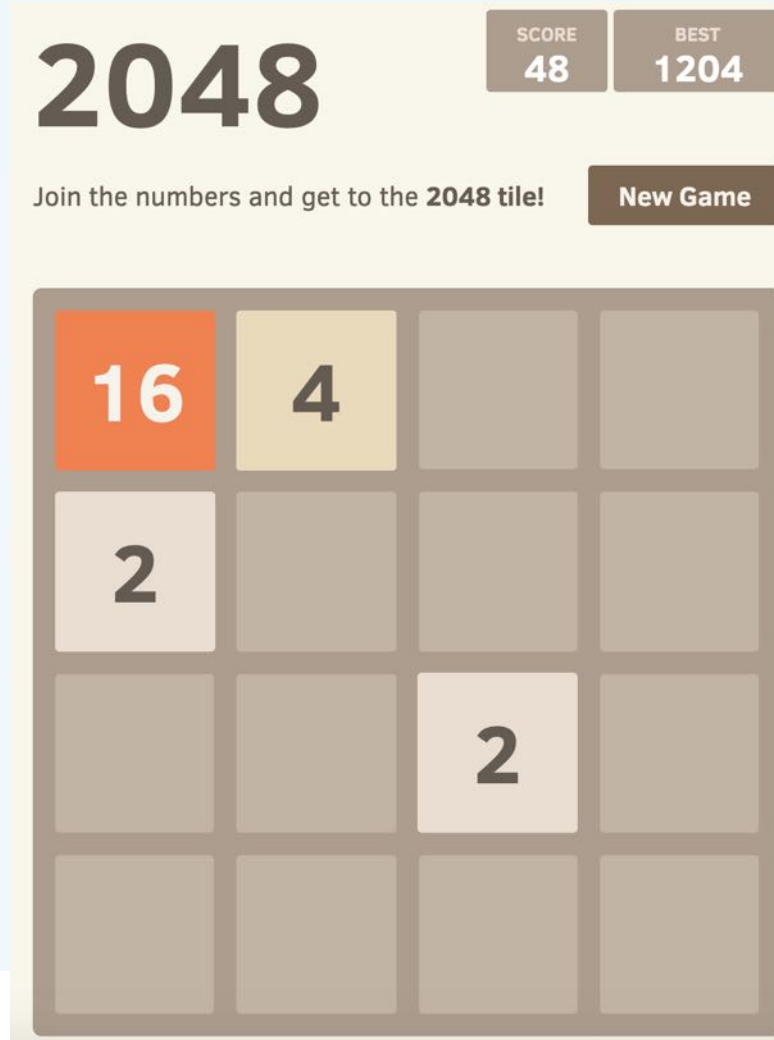
【期末大作业】二〇四八

- › 任务描述
- › 组队
- › 作业评分标准
- › 二〇四八算法竞赛规则
- › 实习作业时间进度
- › 小组算法开发调试指南



游戏原型：2048 (<https://play2048.co/>)

- › 随机**出现** (2)
- › 向某个方向**合并**
- › 相等的数融合
- › 目标：得到最大的数



二〇四八

- › 从原游戏改编为：回合制AI对抗游戏
- › 两队AI分别对应东、西两个相邻的4*4棋盘
- › 轮流**下棋**、**合并**
- › 在边界异色可以同数**吞并**
- › 以一定回合后高分为胜
- › 完全信息决策

	2	32	8				
		8	4				
		4		8			
				32	4	2	

游戏规则简介

A. 基本流程

游戏开始后，每个回合按照四个阶段顺序执行。

- i. 先手方下一个 2 棋子
- ii. 后手方下一个 2 棋子
- iii. 先手方进行一次合并
- iv. 后手方进行一次合并

如果某一阶段某方没有位置可下棋或没有合法的合并方向，那么跳过该阶段。

游戏规则简介

B. 下棋

某方在下棋时需选择其一

- i. 在**对方**的空位中下棋，位置由**己方**决定，棋子归**对方**所有

游戏规则简介

ii. 在**己方**的空位中下棋，位置被**随机**决定，棋子归**己方**所有

需要注意的是，在第二种情况中的随机是由游戏开始时**公开**的随机序列和计算位置的算法计算得到的，即**某方在下棋之前已经知道如果选择在己方下棋，棋子将出现的位置**。在图 3-1 中，对于先手方而言，在所有空位中，绿色框内为可下棋位置，所下棋子归对方所有；蓝色框内为随机决定的可下棋位置，所下棋子归己方所有；红色框内为不可下棋位置。



游戏规则简介

C. 合并

某方选择某方向进行一次**合并**描述的是这样的过程：本方棋子按照**沿该方向靠前者先行**的次序向该方向运动。如果遇到己方空位，将占领空位并继续运动；如果遇到对方空位，不占领空位并停下；如果遇到数值相等的棋子，将吃掉遇到的棋子，占据其位置，并将自身数值修改为原数值的两倍，停下；如果遇到数值不等的棋子，不吃棋并停下。**一个棋子如果吃掉了另一个棋子，在本次合并内不能被吃。**

游戏规则简介

4. 游戏结束

A. 正常结束

当已完成的回合数达到最大回合数时，游戏正常结束，进入计分环节。

B. 意外结束

当出现以下情况时，游戏意外结束。

- i. 某方累计运行时间超出最大时间限制
- ii. 某方运行时发生错误
- iii. 某方产生非法输出

非法输出有以下几类。

- i. 输出结果不合接口要求
- ii. 在不被允许的位置下棋
- iii. 进行无效合并，即合并前后整个棋盘未发生变化

使游戏意外结束的一方将直接判负，进入计分环节。

游戏规则简介

C. 计分与胜负

i. 计分

将双方的棋子按数值高到低的顺序比较个数，一旦某方多即认定该方分数更高。例如，在图 4-2 中，先手方分数更高。

4				2	2	2	2
				2	2	2	2
				2	2	2	2
				2	2	2	2

图 4-2 计分

ii. 胜负

如果某方已经在意外结束时判负，则对方胜出；否则分数高者胜出。

二〇四八：输入输出要求

```
4  # 参赛队伍的AI要求:
5  #
6  # 须写在Player类里
7  #
8  # 须实现两个方法:
9  #
10 # __init__(self, isFirst, array):
11 #     -> 初始化
12 #     -> 参数: isFirst是否先手, 为bool变量, isFirst = True 表示先手
13 #     -> 参数: array随机序列, 为一个长度等于总回合数的list
14 #
15 # output(self, currentRound, board, mode):
16 #     -> 给出己方的决策(下棋的位置或合并的方向)
17 #     -> 参数: currentRound当前轮数, 为从0开始的int
18 #     -> 参数: board棋盘对象
19 #     -> 参数: mode模式, mode = 'position' 对应位置模式, mode = 'direction' 对应方向模式
20 #     -> 返回: 位置模式返回tuple (row, column), row行, 从上到下为0到3的int; column列, 从左到右为0到7的int
21 #     -> 返回: 方向模式返回direction = 0, 1, 2, 3 对应 上, 下, 左, 右
22 #
23 # 其余的属性与方法请自行设计
```


任务描述

- › **编程：依托二〇四八基础设施代码，用Python编写对战算法**
根据当前棋局和回合阶段，返回本方的响应
要求应用本课所学到的数据结构与算法组合，并具有一定的复杂度和智能。
要求代码结构清晰、格式规范、注释丰富。
- › **报告：撰写算法实现过程的实验报告**
包括算法思想阐述、程序代码说明、测试过程报告、小组分工和实验过程总结等4个部分
要求实验报告图文并茂、内容丰富、结构清晰、写作规范、逻辑性强。
- › **竞赛：参加SESSDSA 二〇四八算法竞赛**
与其他小组的算法对战，根据输赢获得竞赛排名
要求对战过程基本无bug、无异常

组队

- › 全班分为F19/N19两个联盟赛区，联盟之间不竞赛
- › 分组原则上每组5人，设组长1名
- › 组队过程由组长确定开始，**确定后**组长开始招募组员
组长确定原则：以自愿报名为主，自愿报名表单见Canvas网站。
- › 组员招募遵循自愿原则，提倡**均衡**原则
- › 组长负责：
 - 召集实习作业过程讨论会
 - 汇总代码和报告
 - 代表小组参加竞赛

作业评分标准

- › 数算实习作业占总评的25%，即**25分**
- › 算法编程占9分
- › 实习报告占8分
- › 联盟内的竞赛排名占8分
参赛无bug无异常得3分；天梯赛出线得5分；季军得6分；亚军得7分；冠军得8分
- › 评分适用于全组同学
每组有额外3分加分，可由组长组织本组民主评议，奖励1~2名表现突出的组员（含组长）。
另外，组长有权对实习过程中表现差的同学提出批评及降分建议。

F19联盟 vs N19联盟

- › 选课一共280人：19级176人；非19级104人
- › 为了均衡实力，缩小小组规模和数量
- › 将全体同学分为19级和非19级两个联盟
- › 联盟内部按照**天梯赛**出线、**淘汰赛**决出冠亚军，获得相应奖励以及神秘奖品
天梯赛和淘汰赛的具体规则待定
- › 最后可以进行联盟之间的冠军友谊赛、挑战赛，以及人机对战表演

实习作业时间进度

- › **组长报名（周四截止），然后正式组队**
- › **即日开始实习作业，开发算法，编程测试，挑战天梯，撰写报告**
注意组员分工明确，协同合作
- › **6月2日（周二）课上进行算法竞赛**
- › **6月9日（周二）前提交完整作业**
包括代码、实验报告

小组算法开发指南

› 详见github代码仓库

github: <https://github.com/pkulab409/sessdsa.2048>

国内码云镜像: <https://gitee.com/chbpku/sessdsa.2048/>

码云镜像国内高速访问, 适合没有梯子的同学

› 各组必看文档

规则文档, 用户文档

› 代码竞技场

项目简介 - 贰零肆捌

课堂对战结果

项目简介

地空数算课2020期末作业：贰零肆捌

- 由经典游戏2048改编而成的算法对抗竞赛
- 可以在 [这里](#) 体验原始规则的游戏
- 项目地址: [sessdsa.2048](#) ; 详细说明请见其文档。

2	32	2	512	2	4	8	2
16	256	16	8		2	8	128
4	32	8	4	2	4	4	16
2	8	4	2	4	2	4	8

匹配对战

自由对战

查看天梯

最近发起的比赛

1/10 下一页 刷新

链接	发起时间	发起方	接收方	对局数	比赛状态
进入	05月11日	 X/EA12小萝莉	 2048.1.1	10/10	已完成
进入	05月11日	 X/EA12小萝莉	 terrible_ai	10/10	已完成

致谢：技术组、内测组

数据结构与算法 (Python)

