

# 数据

## cross.json

正在制作中。

这里保存的是所有的路口，如

```
1  {
2    "total":1,
3    "record":[
4      [0,121.50499,30.83198,"e"]
5    ]
6  }
```

表示路口总数为 1，索引为 0 的路口经纬度为 121.50499,30.83198，这个路口位于 E 教。

目前启用的缩写

- 1,5, 等数字，表示 1 号宿舍楼, 5 号宿舍楼等所在的围合
- dh, 大活
- a,b,c,d,e 教学楼
- p 操场，下图示 1
- lib 图书馆
- r1,r2 一食堂, 二食堂

## 图示

图示 1



图示 2

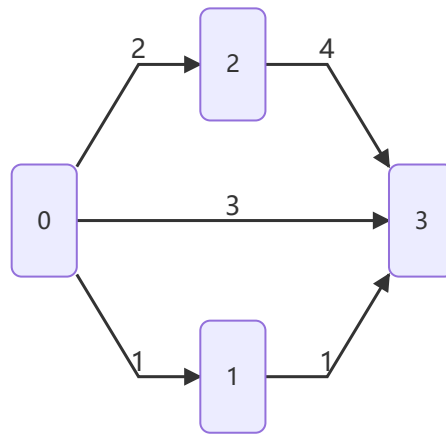


图示 3



## graph.json

这里保存校园的图结构，如这幅图



表示成

```
1  {
2      "vertex": 4,
3      "edge": 5,
4      "adj": [
5          [0, [1, 1], [2, 2], [3, 3]],
6          [1, [3, 1]],
7          [2, [3, 4]],
8          [3]
9      ]
10 }
```

有 4 个顶点和 5 条边，用邻接表描述结构，`[[0,1,1],[0,2,2],[0,3,3]]` 表示从顶点 0 出发有 3 条边

- 0 指向 1，权重为 1
- 0 指向 2，权重为 2
- 0 指向 3，权重为 3

## Python JSON

下面以 `graph.json` 为例，介绍如何用 Python 读取 JSON 文件

```
1  >>> f = open('demo/graph.json', 'r')
2  >>> rows = f.read()
3  >>> f.close()
4  >>> rows
5  '{\n    "vertex": 4,\n    "edge": 5,\n    "adj": [\n        [[0,1,1],[0,2,2],\n        [0,3,3]],\n        [[1,3,1]],\n        [[2,3,4]],\n        []\n    ]\n}'
6  >>> json.loads(rows)
7  {'vertex': 4, 'edge': 5, 'adj': [[[0, 1, 1], [0, 2, 2], [0, 3, 3]], [[1, 3, 1]],\n  [[2, 3, 4]], []]}
8  >>> data = json.loads(rows)
9  >>> data['vertex']
10 4
11 >>> data['edge']
12 5
13 >>> adj = data['adj']
14 >>> adj[0]
15 [[0, 1, 1], [0, 2, 2], [0, 3, 3]]
16 >>> adj[0][0]
```

```

17     [0, 1, 1]
18     >>> food = json.dumps(data)
19     >>> food
20     '{"vertex": 4, "edge": 5, "adj": [[[0, 1, 1], [0, 2, 2], [0, 3, 3]], [[1, 3, 1]],
    [[2, 3, 4]], []]}'

```

前后端就是用 JSON 交换数据的。

```

1     >>> import PyerSP as psp
2     >>> g = psp.graph('demo/graph.json')
3     >>> g.path(1,3)
4     '{"dist":0.8, "path":[1,2,4,3]}'
5     // 表示从 1 到 3 最短的路径是 1→2→4→3，总长度是 0.8

```

或者这样子

```

1     # PyerSP.py
2     import json
3     ...
4     class Path(object):
5         def __init__(self,dist,path):
6             self.dist = dist
7             self.path = path
8         def __str__(self):
9             return json.dumps({'dist':self.dist, 'path':self.path})

```