lec1_step5

October 14, 2020

In [ ]:
```
## Python basics for novice data scientists, supported by Wagatsuma Lab@Kyutech
#
# The MIT License (MIT): Copyright (c) 2020 Hiroaki Wagatsuma and Wagatsuma Lab@Kyutec
#
# Permission is hereby granted, free of charge, to any person obtaining a copy of this
# The above copyright notice and this permission notice shall be included in all copie
# THE SOFTWARE IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED,
#
# # @Time    : 2020-10-14
# # @Author  : Hiroaki Wagatsuma
# # @Site    : https://github.com/hirowgit/2A_python_basic_course
# # @IDE     : Python 3.7.7 (default, Mar 10 2020, 15:43:27) [Clang 10.0.0 (clang-1000
# # @File    : lec1_step5.py
```

In [ ]:
```
# Practice 2-2 (page 21/28)
# https://www.slideshare.net/tadahirotaniguchi0624/2-46861654
```

In [8]:
```
# open list and closed list
```

In [10]:
```
# first idea
OpenList=[1,2,3,4]
```

In [5]:
```
OpenList[1]
```

Out[5]: 2

In [6]:
```
OpenList[0]  # note array start from [0] like C, C++
```

Out[6]: 1

In [105]:
```
# As you see in Fig 2.9, open list and closed list should be defined at each node.
# Therefore those lists require multiple open and closed lists for each node.
# It implies dictionary is a good option.
TargetGraph={
    'S':'A','B',
    'A':'S','C','D',
    'B':'S','C',
    'C':'A','B','D',
    'D':'A','C',
#    'G':'unknown now
}
```

```
  File "<ipython-input-105-1bf0c221c17f>", line 5
    'S':'A','B',
            ^
SyntaxError: invalid syntax
```

```
In [110]: TargetGraph={
              'S':['A','B'],
              'A':['S','C','D'],
              'B':['S','C'],
              'C':['A','B','D'],
              'D':['A','C']
          #    'G':'unknown now
          }
```

```
In [111]: TargetGraph['S']
```

```
Out[111]: ['A', 'B']
```

```
In [112]: TargetGraph['S'][0]
```

```
Out[112]: 'A'
```

```
In [113]: TargetGraph['S'].append("G")
```

```
In [23]: print(TargetGraph)
```

```
{'S': ['A', 'B', 'G'], 'A': ['S', 'B'], 'B': ['A', 'B'], 'C': ['A', 'B'], 'D': ['A', 'B']}
```

```
In [114]: # If you want to delete the last item
          del TargetGraph['S'][-1]
          print(TargetGraph)
```

```
{'S': ['A', 'B'], 'A': ['S', 'C', 'D'], 'B': ['S', 'C'], 'C': ['A', 'B', 'D'], 'D': ['A', 'C']}
```

```
In [115]: tList=[]
          if tList:
              print('Not Empty')
          else:
              print('Empty')
```

```
Empty
```

```
In [116]: tList=[1,2,3,4,5]
          while tList:
              del tList[0]
              print(tList)
          print('completed')
```

```
[2, 3, 4, 5]
[3, 4, 5]
[4, 5]
[5]
[]
completed
```

```
In [117]: OpenList=['S']
          OpenList.insert(0,['A','B'])
          print(OpenList)
```

```
[['A', 'B'], 'S']
```

```
In [118]: sList=['A','B']
          [d for d in sList]
```

```
Out[118]: ['A', 'B']
```

```
In [119]: TargetGraph['A']
```

```
Out[119]: ['S', 'C', 'D']
```

```
In [126]: OpenList=['S']
          sList=['A','B']
          OpenList.insert(0, sList)
          OpenList=[d for d in OpenList]
          print(OpenList)
          OpenList=[item for i in OpenList for item in i]
          print(OpenList)
```

```
[['A', 'B'], 'S']
['A', 'B', 'S']
```

```
In [78]: if 'A' in ['A', 'B', 'S']:
             print('Yes')
```

```
Yes
```

```
In [79]: if 'A' not in ['A', 'B', 'S']:
             print('Yes')
```

```
In [88]: tList=[]
         addList=['A', 'B', 'S']
         ClosedList=['S']
         activeNode=[item for item in addList if item not in ClosedList]
         activeNode
```

```
Out[88]: ['A', 'B']
```

```
In [134]: OpenList=['S']
          state='S'
          OpenList.insert(0, TargetGraph[state])
          print(OpenList)

          OpenList=['S']
          ClosedList=['S']
          state='S'
          print(TargetGraph[state])
          activeNodes=[item for item in TargetGraph[state] if item not in ClosedList]
          OpenList.insert(0, activeNodes)
          OpenList=[item for i in OpenList for item in i if item not in ClosedList]
          print(OpenList)
```

```
[['A', 'B'], 'S']
['A', 'B']
['A', 'B']
```

```
In [135]: OpenList=['S']
          ClosedList=[]
          while OpenList:
              state=OpenList[0]
              del OpenList[0]
              ClosedList.append(state)
              print(state)
              if state=='G':
                  break
           #   activeNodes=TargetGraph[state]
              activeNodes=[item for item in TargetGraph[state] if item not in ClosedList]
              OpenList.insert(0, activeNodes)
          #    OpenList=[item for i in OpenList for item in i]
              OpenList=[item for i in OpenList for item in i if item not in ClosedList]
          print('completed')
```

```
S
A
C
B
D
completed
```

```
In [136]: TargetGraph={
              'A':['B','C'],
              'B':['A','D','E'],
              'C':['A','F','G','H'],
```

```
            'D':['B','I'],
            'E':['B'],
            'F':['C'],
            'G':['C','J'],
            'H':['C'],
            'I':['D'],
            'J':['G']
        #     'G':'unknown now
        }
```

In [143]:
```
OpenList=['A']
ClosedList=[]
k=1
while OpenList:
    state=OpenList[0]
    del OpenList[0]
    ClosedList.append(state)
    print(str(k)+": "+state)
    if state=='Goal':
        break
 #   activeNodes=TargetGraph[state]
    activeNodes=[item for item in TargetGraph[state] if item not in ClosedList]
    OpenList.insert(0, activeNodes)
#    OpenList=[item for i in OpenList for item in i]
    OpenList=[item for i in OpenList for item in i if item not in ClosedList]
    k=k+1
print('completed')
```

```
1: A
2: B
3: D
4: I
5: E
6: C
7: F
8: G
9: J
10: H
completed
```

In [ ]: