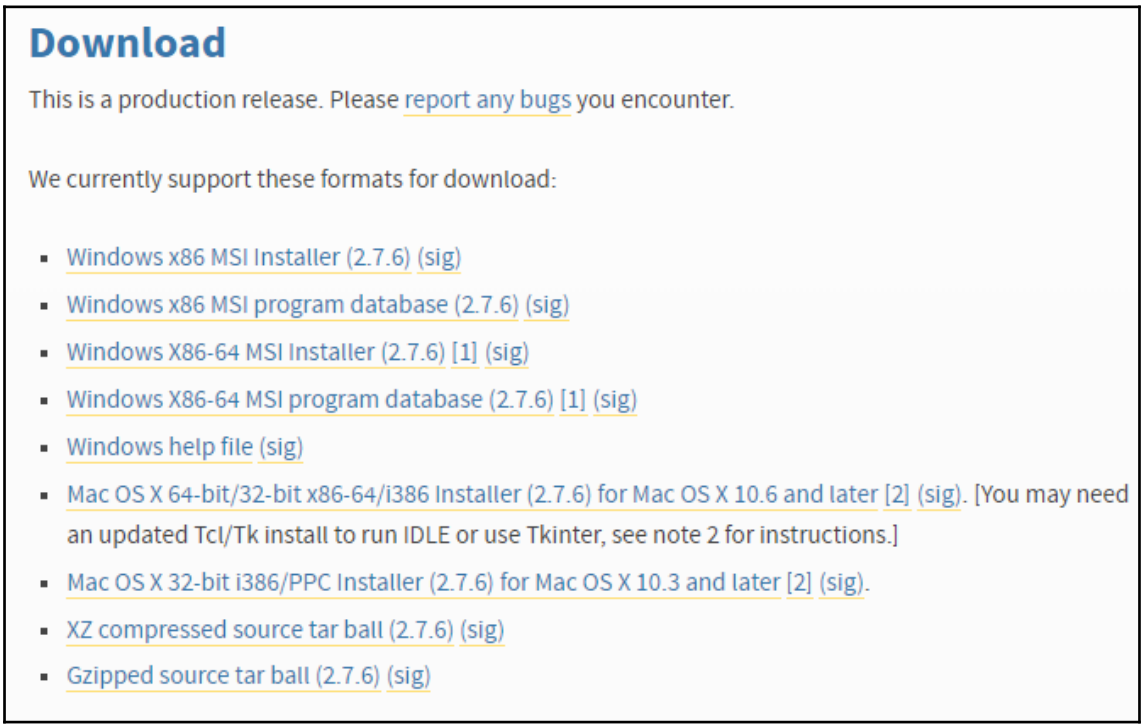
Chapter 1: Understanding the Depth-First Search Algorithm

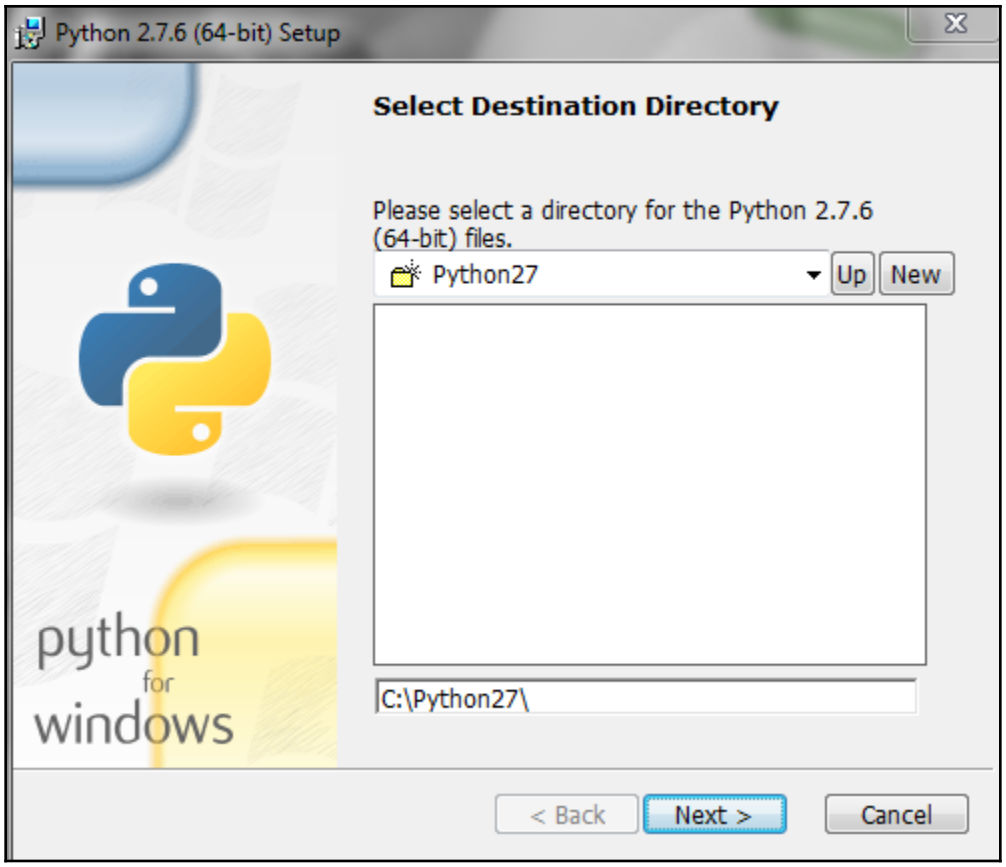


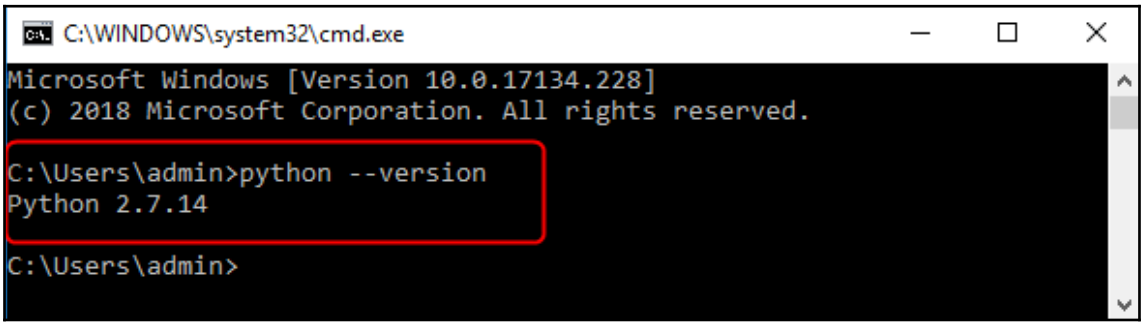
Download

This is a production release. Please report any bugs you encounter.

We currently support these formats for download:

- Windows x86 MSI Installer (2.7.6) (sig)
- Windows x86 MSI program database (2.7.6) (sig)
- Windows X86-64 MSI Installer (2.7.6) [1] (sig)
- Windows X86-64 MSI program database (2.7.6) [1] (sig)
- Windows help file (sig)
- Mac OS X 64-bit/32-bit x86-64/i386 Installer (2.7.6) for Mac OS X 10.6 and later [2] (sig). [You may need an updated Tcl/Tk install to run IDLE or use Tkinter, see note 2 for instructions.]
- Mac OS X 32-bit i386/PPC Installer (2.7.6) for Mac OS X 10.3 and later [2] (sig).
- XZ compressed source tar ball (2.7.6) (sig)
- Gzipped source tar ball (2.7.6) (sig)


Python 2.7.6 (64-bit) Setup
Select Destination Directory
Please select a directory for the Python 2.7.6 (64-bit) files.
Python27
Up
New
C:\Python27\
python
for
windows
< Back
Next >
Cancel


C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.228]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\admin>python --version
Python 2.7.14
C:\Users\admin>

- **Stable and development rpms for Redhat Enterprise, or Centos systems*** available but are out of date.

Windows

- **Development Windows install packages**
- **Stable 2.38 Windows install packages**
- **Cygwin Ports*** provides a port of Graphviz to Cygwin.
- **WinGraphviz*** Win32/COM object (dot/neato library for Visual Basic and ASP).

Mostly correct notes for building Graphviz on Windows can be found **here**.

About Download Gallery Documentation Theory and Publications License
Resources Credits FAQ Contact Twitter Issues/Bugs

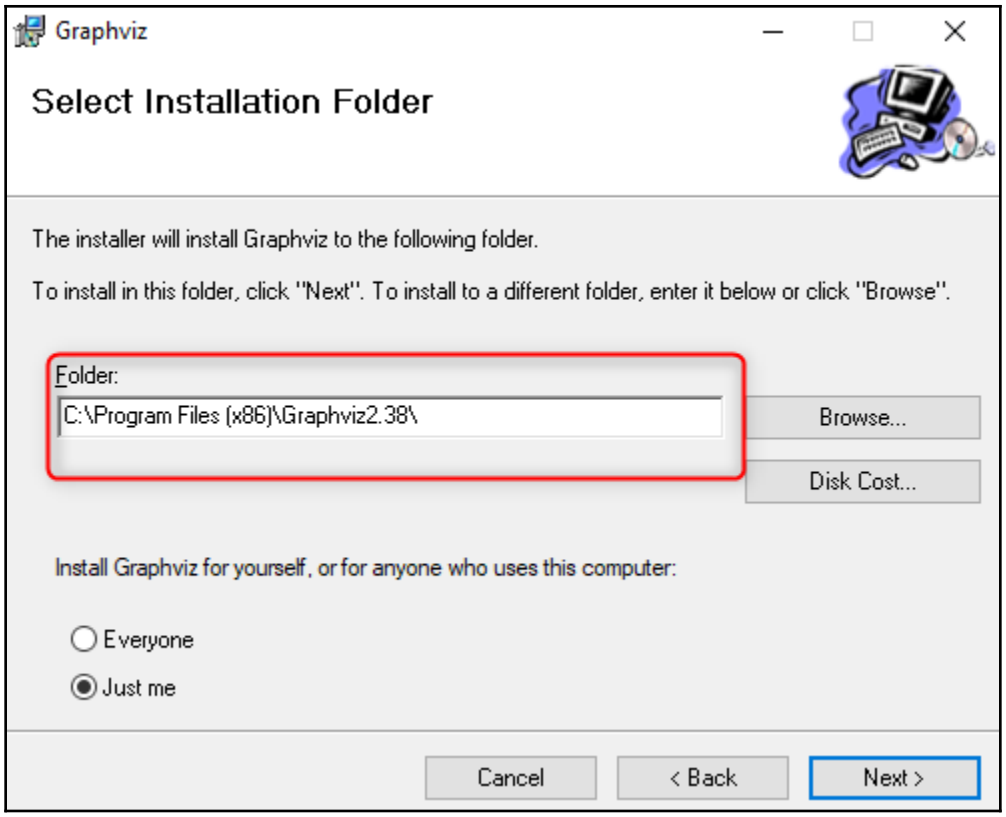
Graphviz - Graph Visualization Software

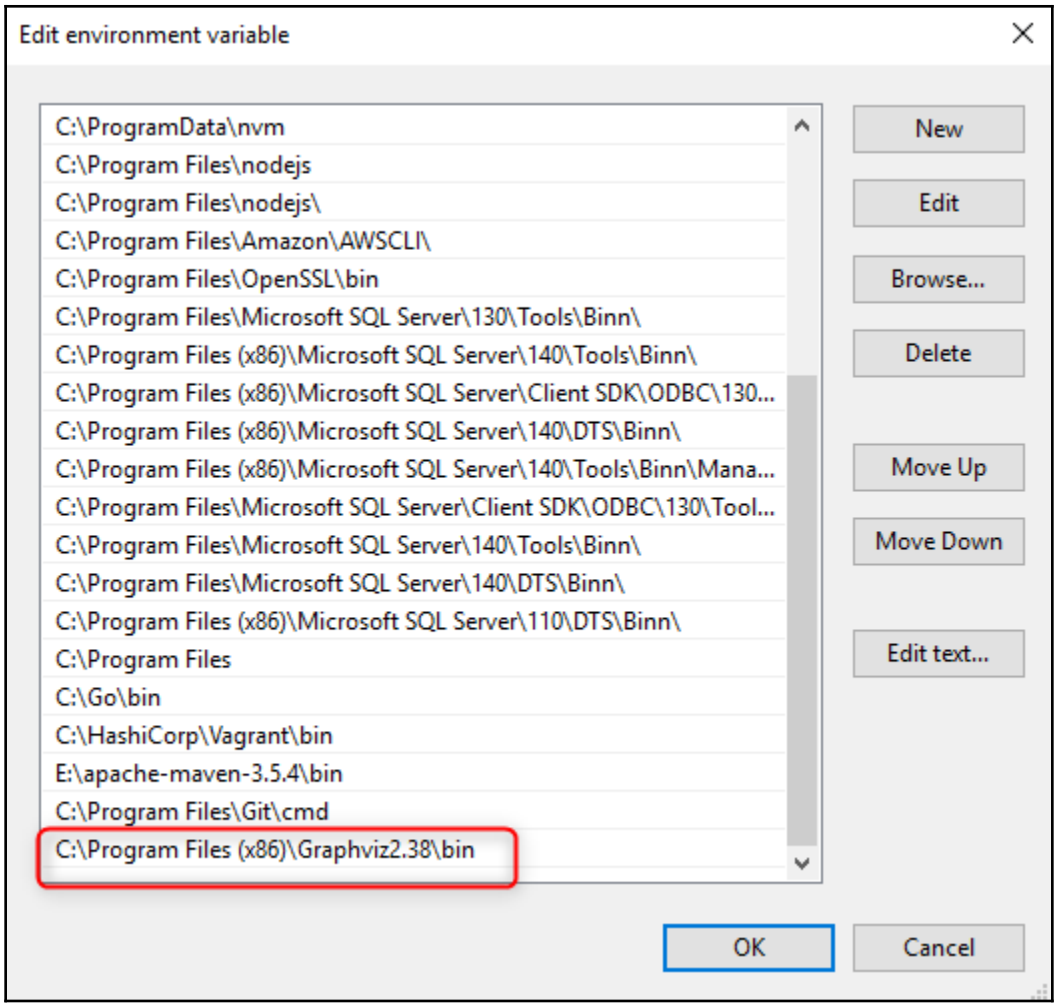
Windows Packages

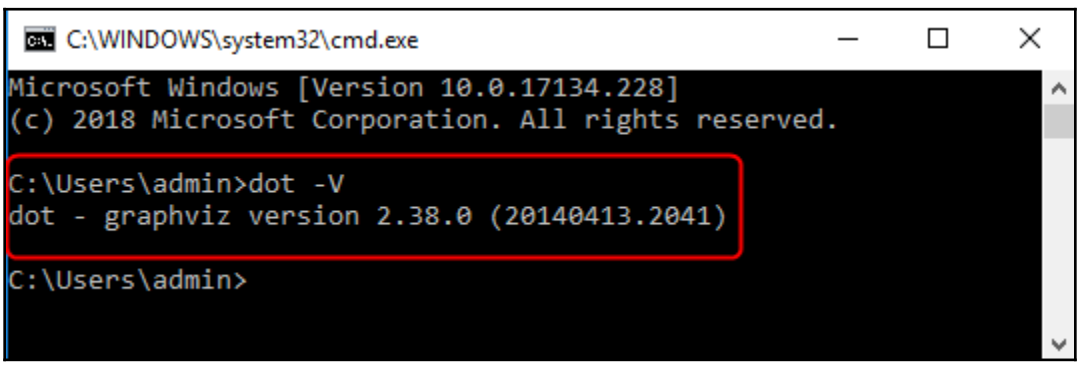
Note: These Visual Studio packages do not alter the PATH variable or access the registry at all. If you wish to use the command-line interface to Graphviz or are using some other program that calls a Graphviz program, you will need to set the PATH variable yourself.

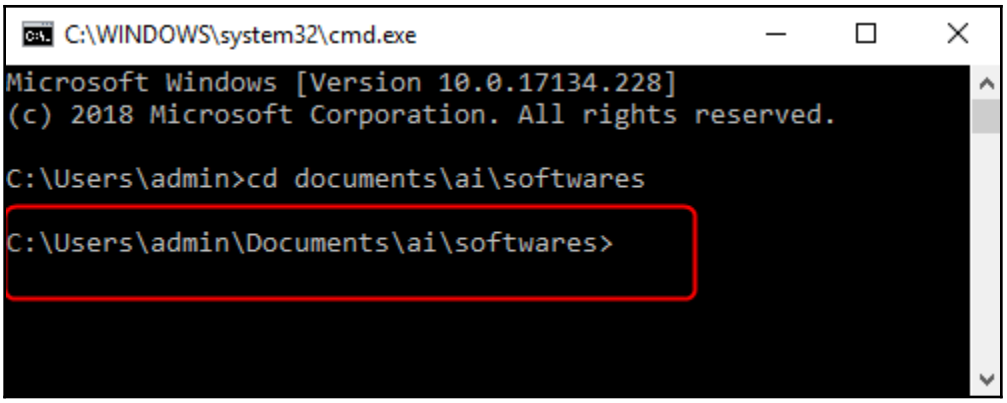
2.38 Stable Release

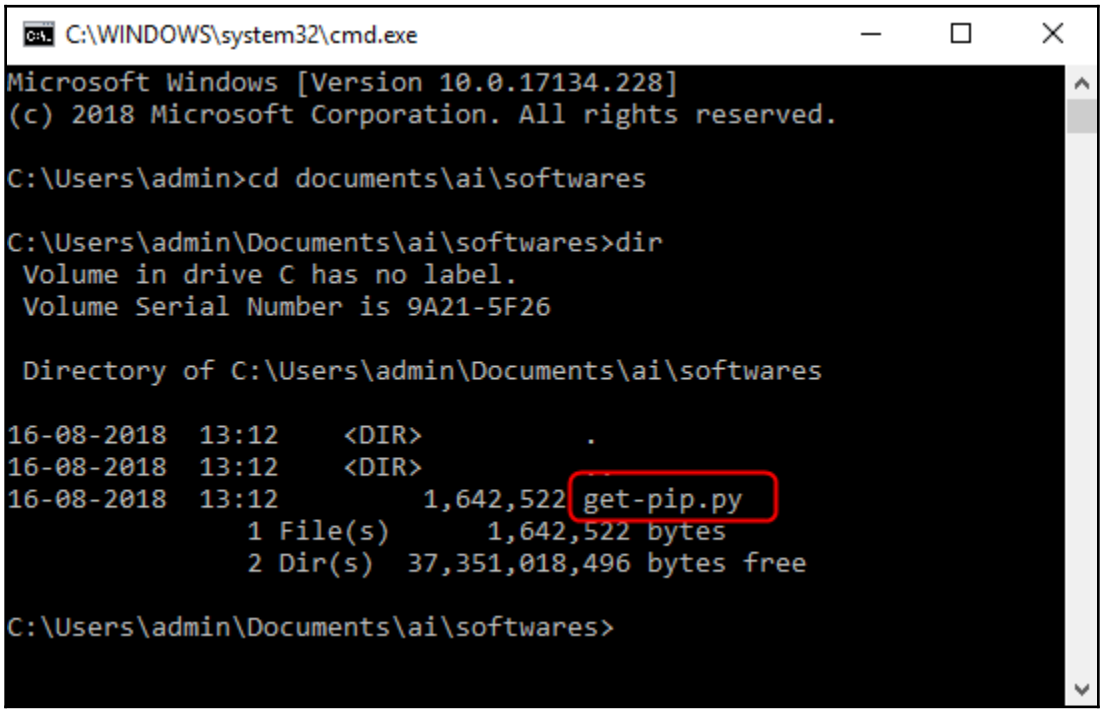
- **graphviz-2.38.msi**
- **graphviz-2.38.zip**

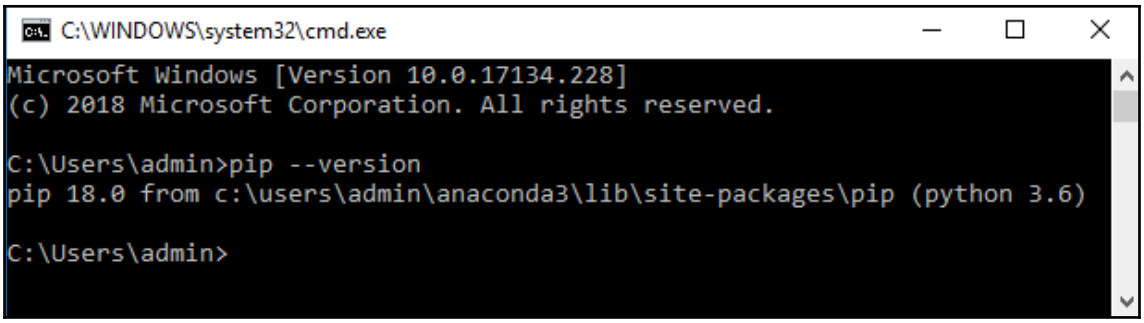

Graphviz
Select Installation Folder
The installer will install Graphviz to the following folder.
To install in this folder, click "Next". To install to a different folder, enter it below or click "Browse".
Folder:
C:\Program Files (x86)\Graphviz2.38\
Browse...
Disk Cost...
Install Graphviz for yourself, or for anyone who uses this computer:
Everyone
Just me
Cancel
< Back
Next >

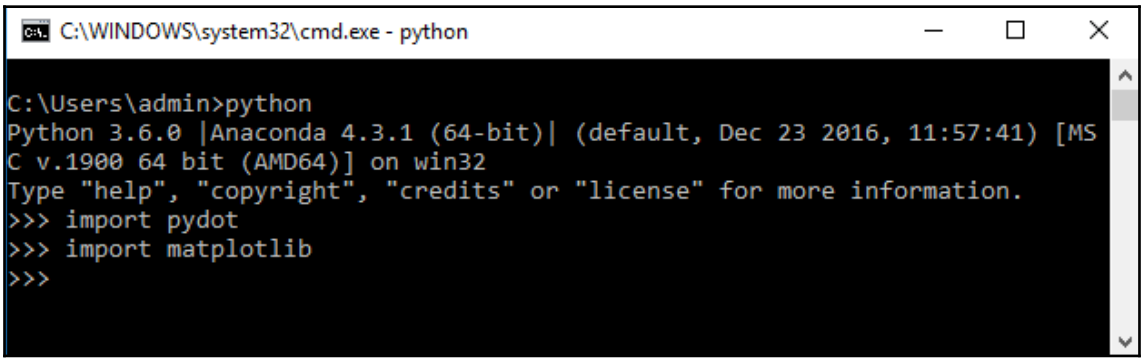

Edit environment variable
C:\ProgramData\nvm
C:\Program Files\nodejs
C:\Program Files\nodejs\
C:\Program Files\Amazon\AWSCLI\
C:\Program Files\OpenSSL\bin
C:\Program Files\Microsoft SQL Server\130\Tools\Binn\
C:\Program Files (x86)\Microsoft SQL Server\140\Tools\Binn\
C:\Program Files (x86)\Microsoft SQL Server\Client SDK\ODBC\130...
C:\Program Files (x86)\Microsoft SQL Server\140\DTS\Binn\
C:\Program Files (x86)\Microsoft SQL Server\140\Tools\Binn\Mana...
C:\Program Files\Microsoft SQL Server\Client SDK\ODBC\130\Tool...
C:\Program Files\Microsoft SQL Server\140\Tools\Binn\
C:\Program Files\Microsoft SQL Server\140\DTS\Binn\
C:\Program Files (x86)\Microsoft SQL Server\110\DTS\Binn\
C:\Program Files
C:\Go\bin
C:\HashiCorp\Vagrant\bin
E:\apache-maven-3.5.4\bin
C:\Program Files\Git\cmd
C:\Program Files (x86)\Graphviz2.38\bin
New
Edit
Browse...
Delete
Move Up
Move Down
Edit text...
OK
Cancel

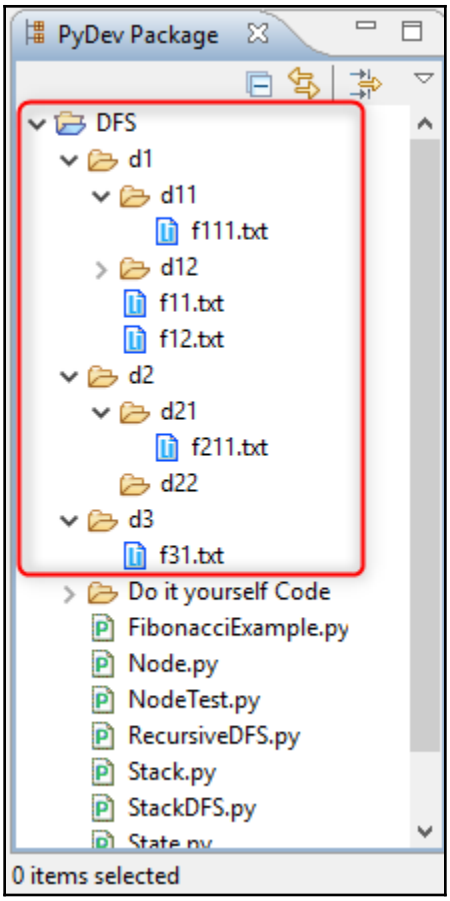

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.228]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\admin>dot -V
dot - graphviz version 2.38.0 (20140413.2041)
C:\Users\admin>


C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.228]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\admin>cd documents\ai\softwares
C:\Users\admin\Documents\ai\softwares>

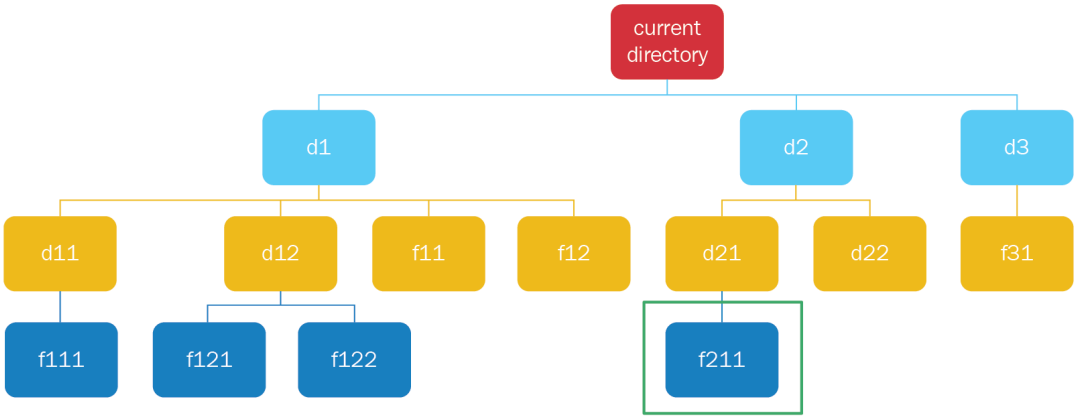

C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.228]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\admin>cd documents\ai\softwares
C:\Users\admin\Documents\ai\softwares>dir
Volume in drive C has no label.
Volume Serial Number is 9A21-5F26
Directory of C:\Users\admin\Documents\ai\softwares
16-08-2018 13:12 <DIR> .
16-08-2018 13:12 <DIR> ..
16-08-2018 13:12 1,642,522 get-pip.py
1 File(s) 1,642,522 bytes
2 Dir(s) 37,351,018,496 bytes free
C:\Users\admin\Documents\ai\softwares>


C:\WINDOWS\system32\cmd.exe
Microsoft Windows [Version 10.0.17134.228]
(c) 2018 Microsoft Corporation. All rights reserved.
C:\Users\admin>pip --version
pip 18.0 from c:\users\admin\anaconda3\lib\site-packages\pip (python 3.6)
C:\Users\admin>


C:\WINDOWS\system32\cmd.exe - python
C:\Users\admin>python
Python 3.6.0 |Anaconda 4.3.1 (64-bit)| (default, Dec 23 2016, 11:57:41) [MS
C v.1900 64 bit (AMD64)] on win32
Type "help", "copyright", "credits" or "license" for more information.
>>> import pydot
>>> import matplotlib
>>>


PyDev Package
DFS
d1
d11
f111.txt
d12
f11.txt
f12.txt
d2
d21
f211.txt
d22
d3
f31.txt
Do it yourself Code
FibonacciExample.py
Node.py
NodeTest.py
RecursiveDFS.py
Stack.py
StackDFS.py
0 items selected

Tree Representation of File System



Ingredients of Searching

Initial State

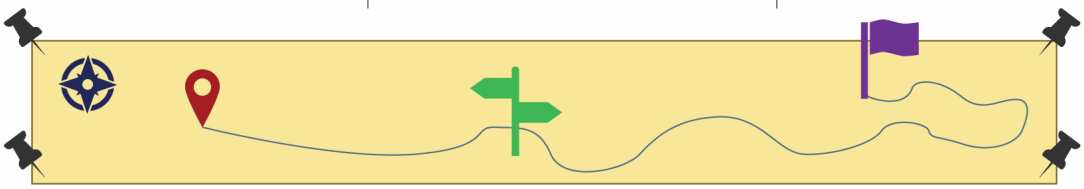
Where do we start searching from?

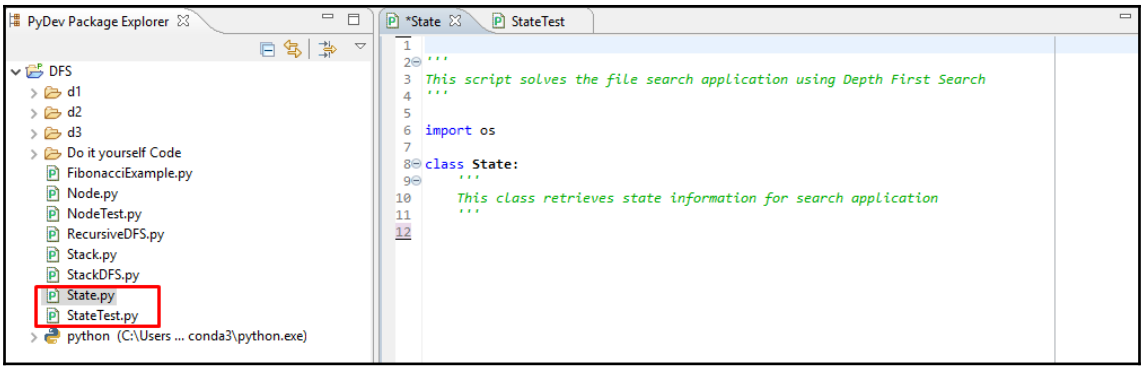
Successor Function

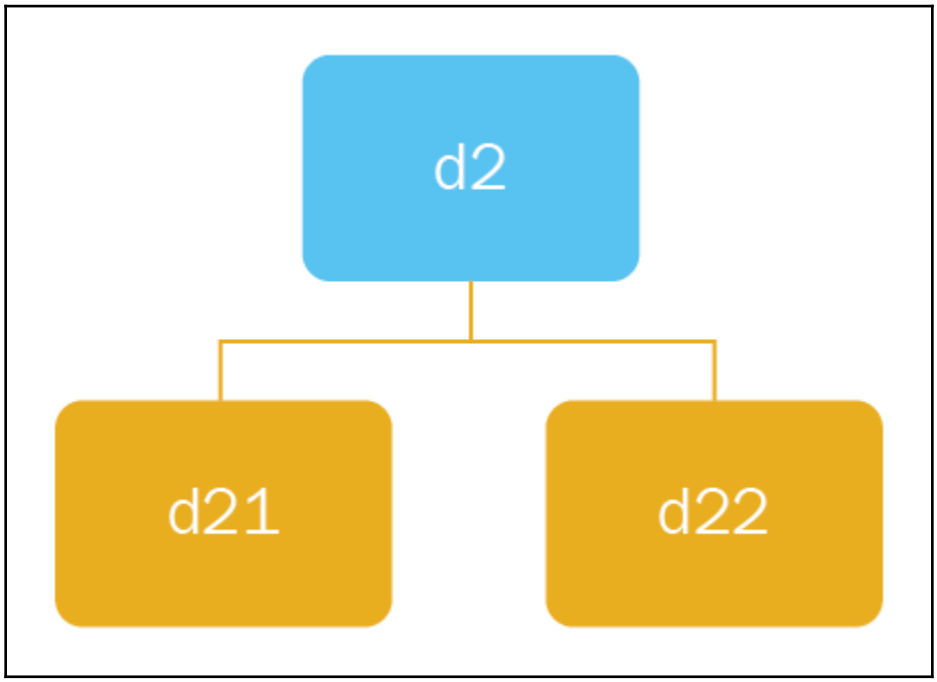
How do we explore from current state?

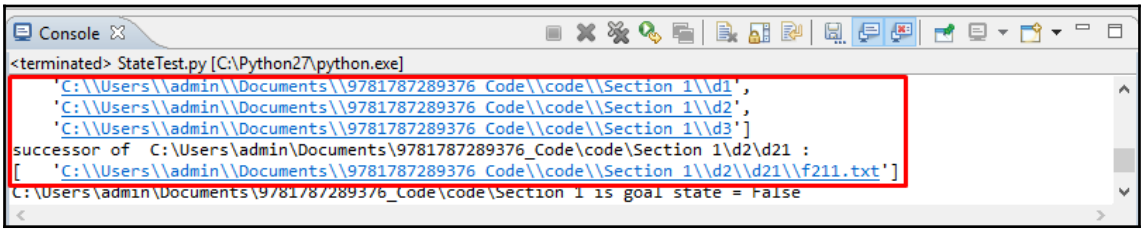
Goal Function

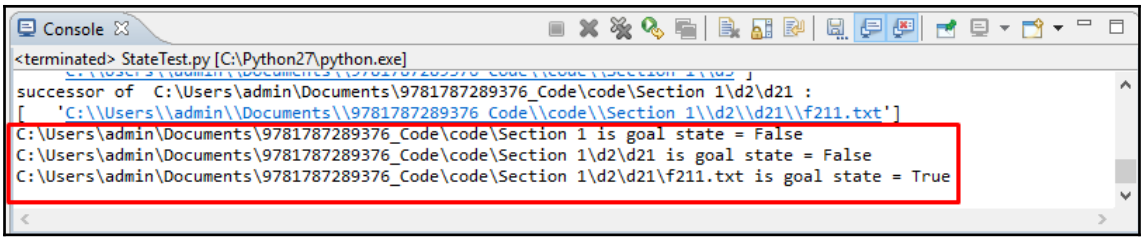
How do we know we found the solution?

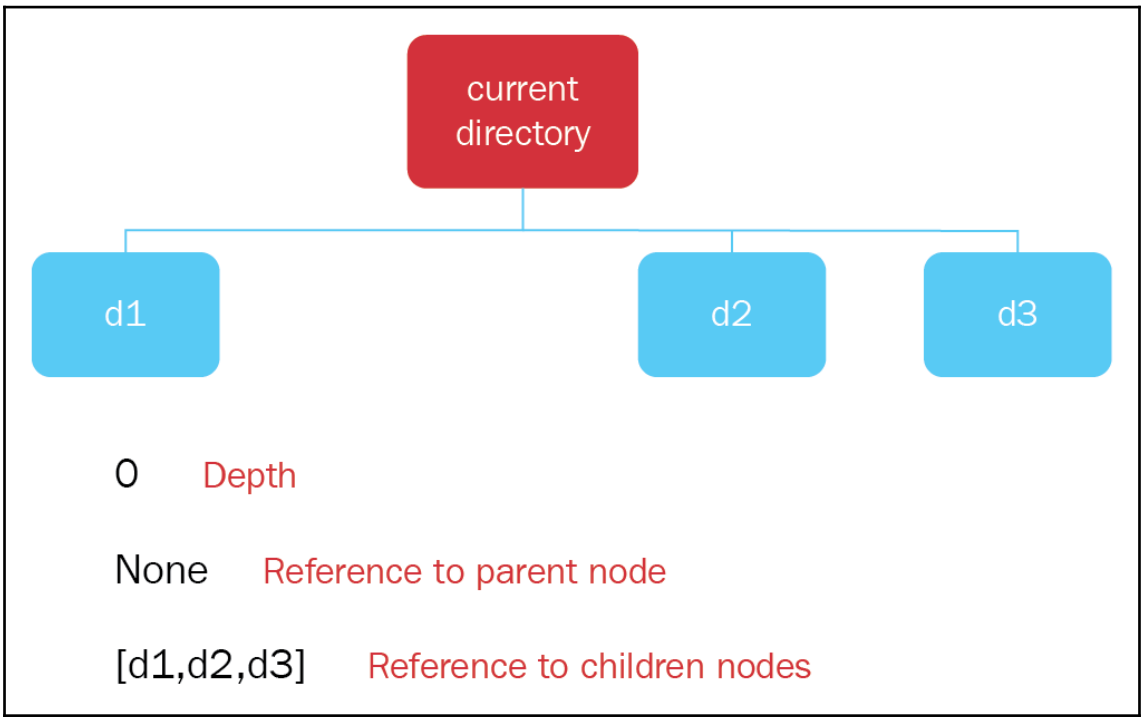


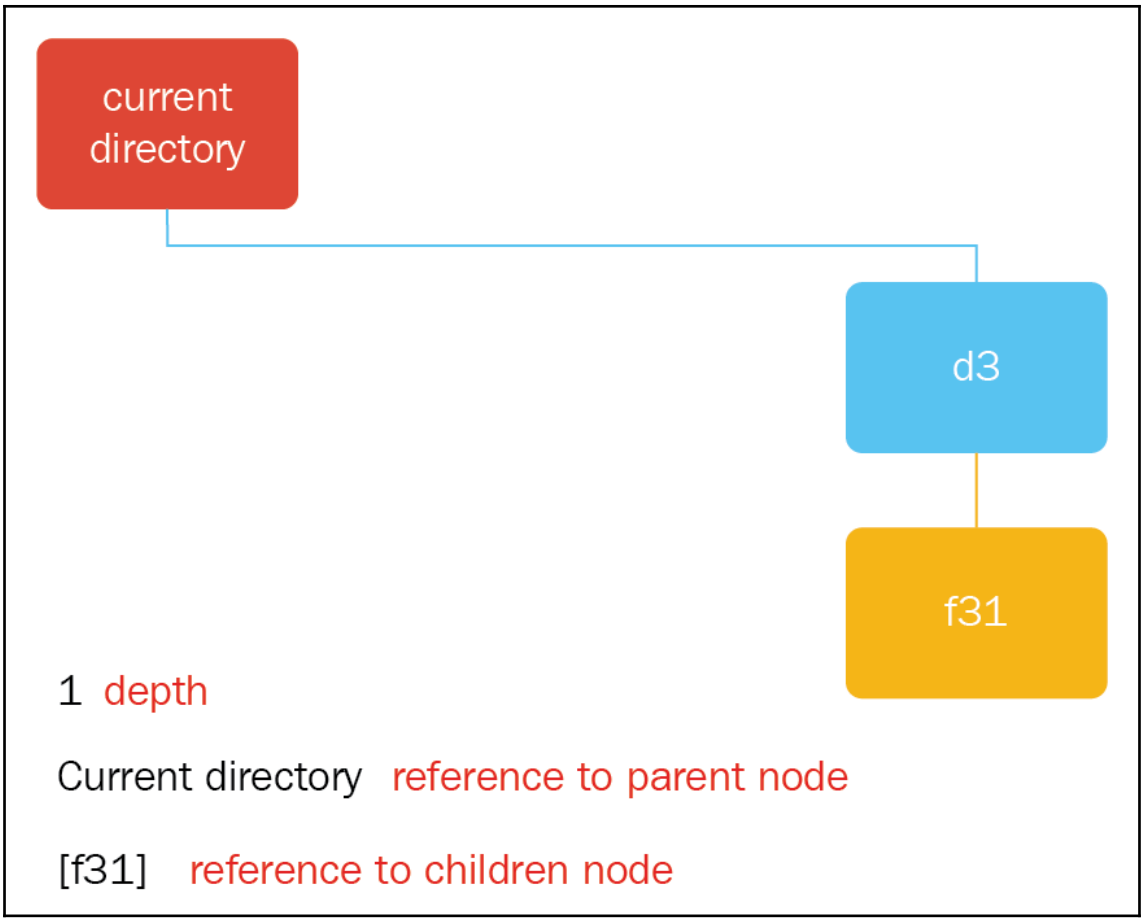

PyDev Package Explorer
DFS
d1
d2
d3
Do it yourself Code
FibonacciExample.py
Node.py
NodeTest.py
RecursiveDFS.py
Stack.py
StackDFS.py
State.py
StateTest.py
python (C:\Users ... conda3\python.exe)
*State
StateTest
'''
This script solves the file search application using Depth First Search
'''
import os
class State:
'''
This class retrieves state information for search application
'''

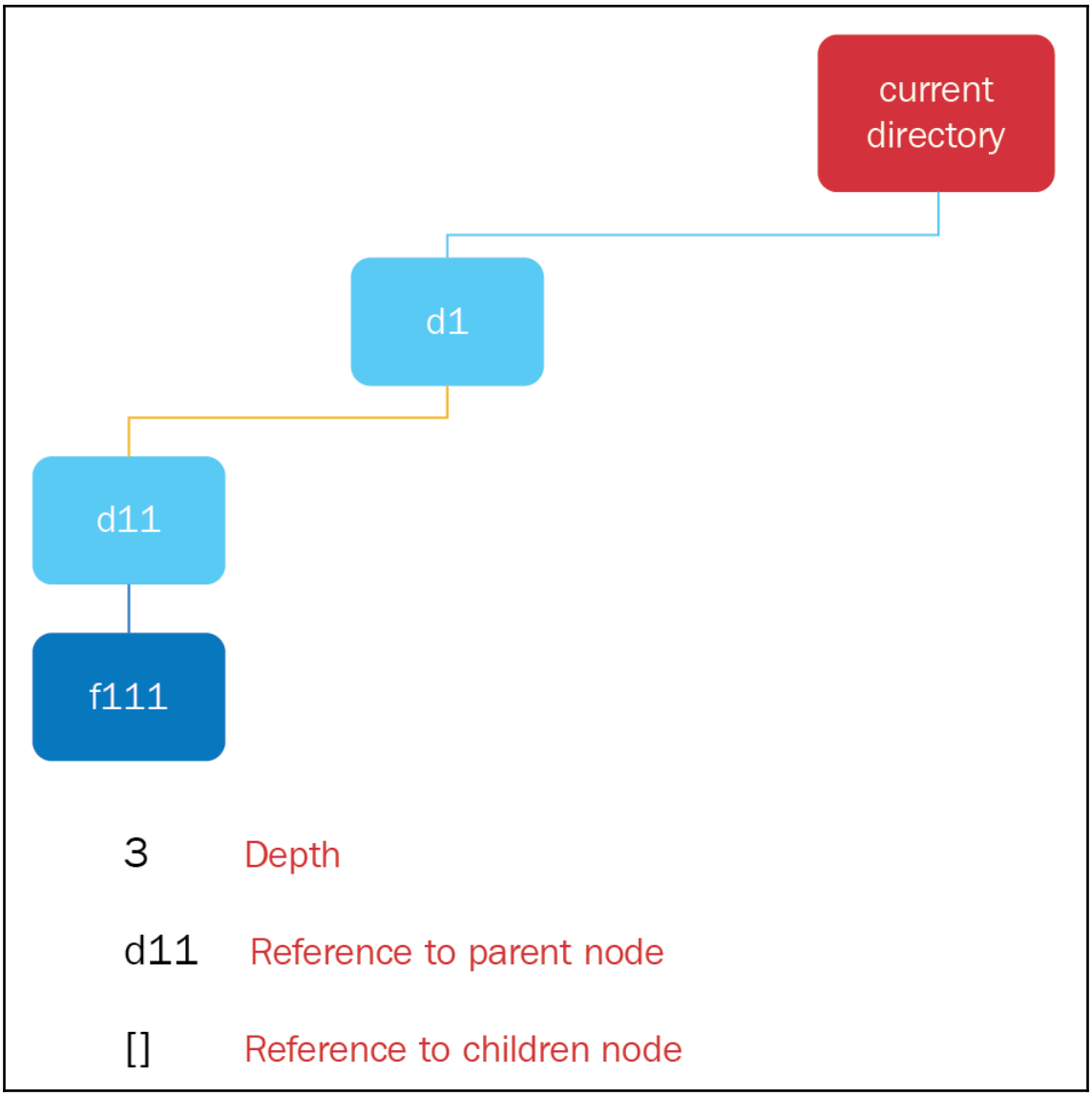

d2
d21
d22


Console
<terminated> StateTest.py [C:\Python27\python.exe]
'C:\\Users\\admin\\Documents\\9781787289376_Code\\code\\Section 1\\d1',
'C:\\Users\\admin\\Documents\\9781787289376_Code\\code\\Section 1\\d2',
'C:\\Users\\admin\\Documents\\9781787289376_Code\\code\\Section 1\\d3']
successor of C:\Users\admin\Documents\9781787289376_Code\code\Section 1\d2\d21 :
['C:\\Users\\admin\\Documents\\9781787289376_Code\\code\\Section 1\\d2\\d21\\f211.txt']
C:\Users\admin\Documents\9781787289376_Code\code\Section 1 is goal state = False


Console
<terminated> StateTest.py [C:\Python27\python.exe]
successor of C:\Users\admin\Documents\9781787289376_Code\code\Section 1\d2\d21 :
['C:\\Users\\admin\\Documents\\9781787289376_Code\\code\\Section 1\\d2\\d21\\f211.txt']
C:\Users\admin\Documents\9781787289376_Code\code\Section 1 is goal state = False
C:\Users\admin\Documents\9781787289376_Code\code\Section 1\d2\d21 is goal state = False
C:\Users\admin\Documents\9781787289376_Code\code\Section 1\d2\d21\f211.txt is goal state = True


current directory
d1
d2
d3
0 Depth
None Reference to parent node
[d1,d2,d3] Reference to children nodes


current directory
d3
f31
1 depth
Current directory reference to parent node
[f31] reference to children node

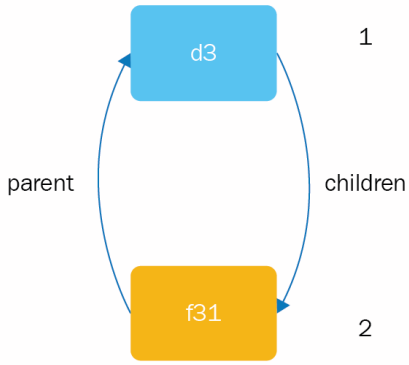

current directory
d1
d11
f111
3 Depth
d11 Reference to parent node
[] Reference to children node

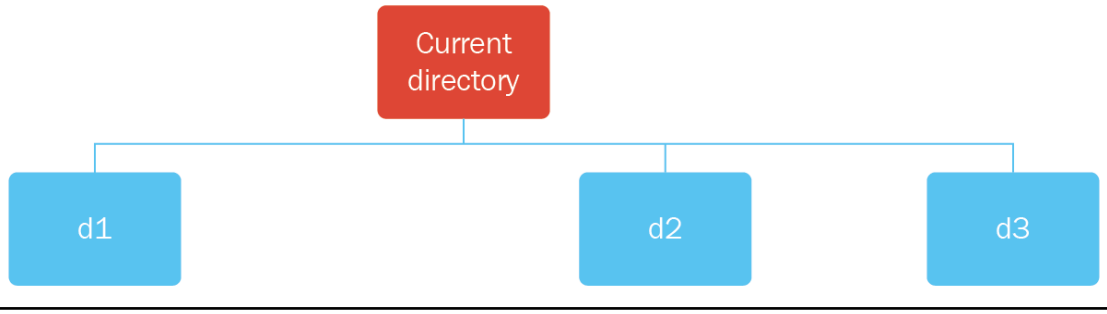
How to Add a Child Node?

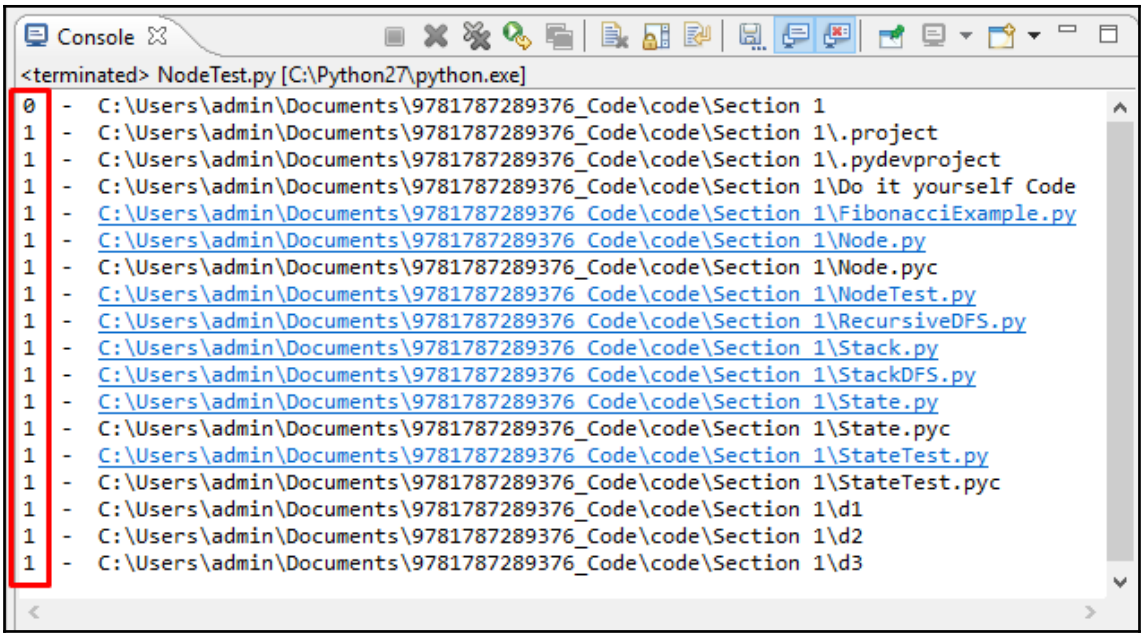
ParentNode.children.add(ChildNode)

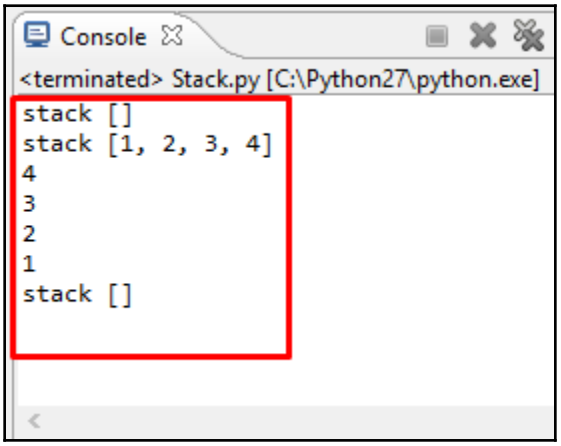
ChildNode.parent = ParentNode

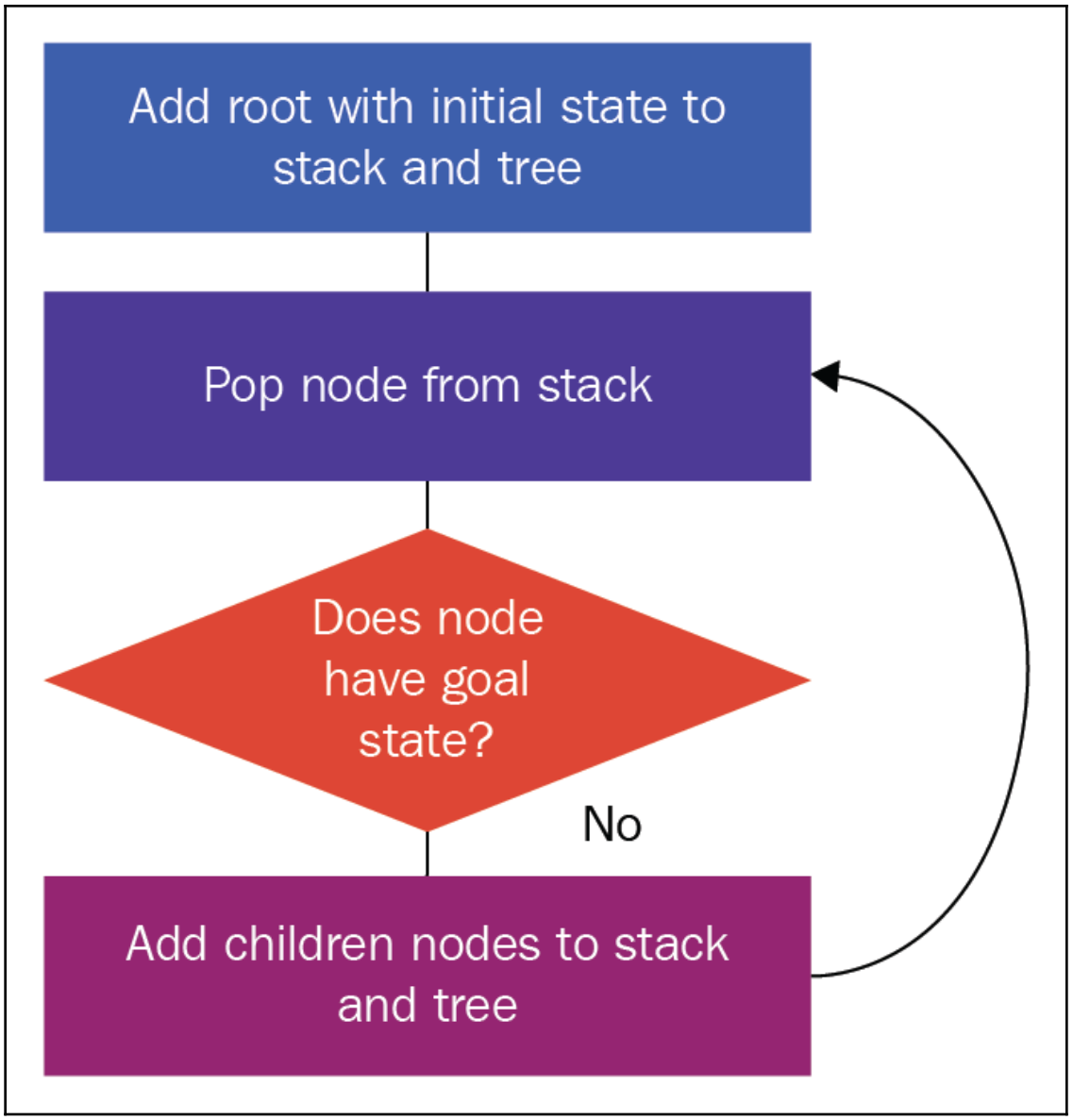
ChildNode.depth = ParentNode.depth + 1

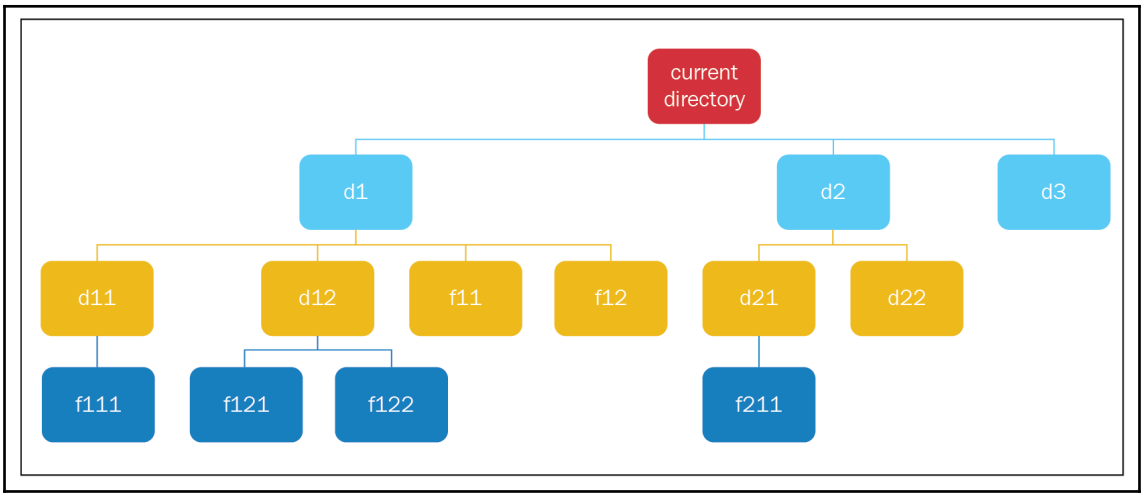





Console
<terminated> NodeTest.py [C:\Python27\python.exe]
0 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\.project
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\.pydevproject
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\FibonacciExample.py
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Node.py
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Node.pyc
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\NodeTest.py
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\RecursiveDFS.py
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Stack.py
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\StackDFS.py
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\State.py
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\State.pyc
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\StateTest.py
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\StateTest.pyc
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\d1
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\d2
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\d3


Console
<terminated> Stack.py [C:\Python27\python.exe]
stack []
stack [1, 2, 3, 4]
4
3
2
1
stack []

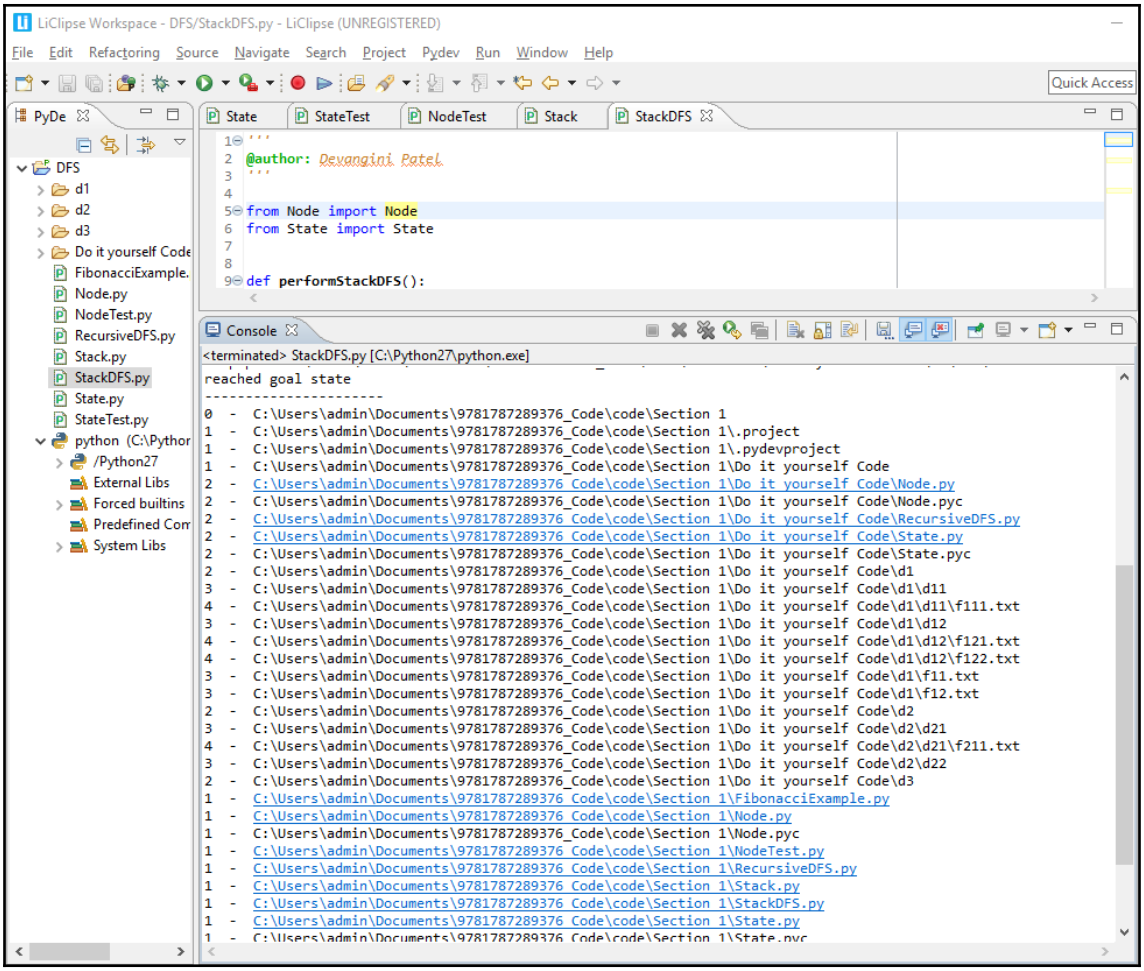

Add root with initial state to stack and tree
Pop node from stack
Does node have goal state?
No
Add children nodes to stack and tree

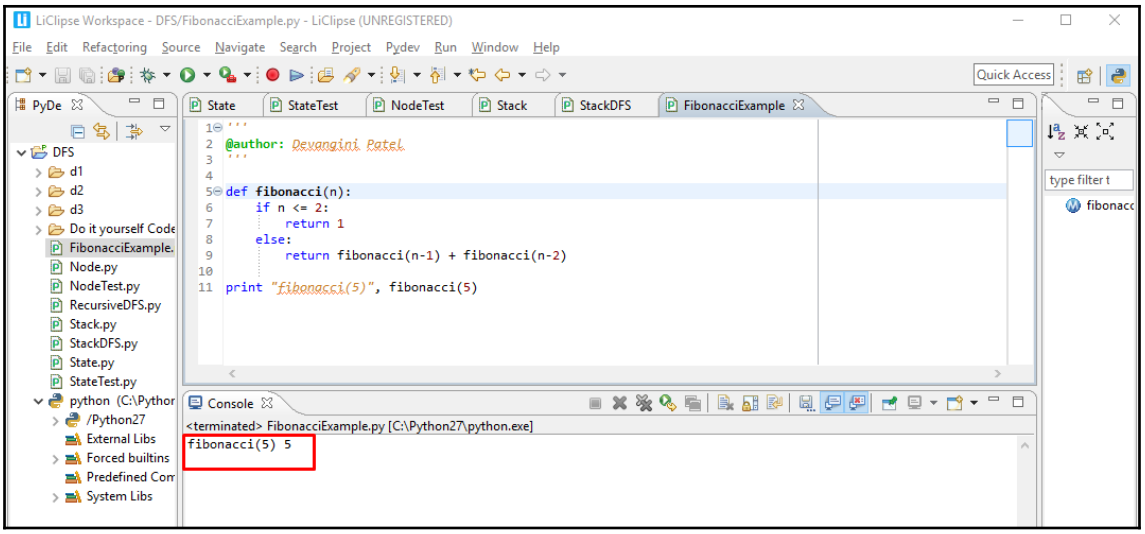


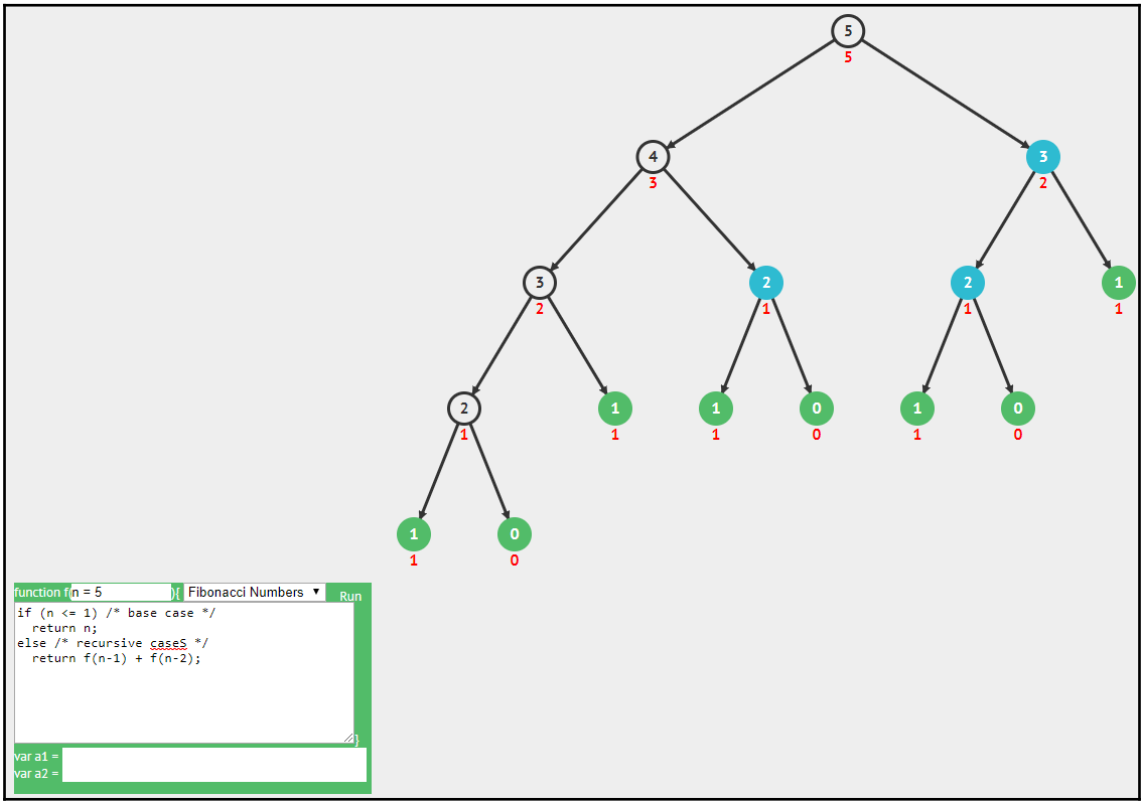
Console

<terminated> StackDFS.py [C:\Python27\python.exe]

```
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\.project
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\.pydevproject
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\Node.py
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\Node.pyc
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\RecursiveDFS.py
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\State.py
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\State.pyc
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d11
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d11\f111.txt
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d12
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d12\f121.txt
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d12\f122.txt
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\f11.txt
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\f12.txt
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d2
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d2\d21
-- pop -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d2\d21\f211.txt
reached goal state
----------------------
```

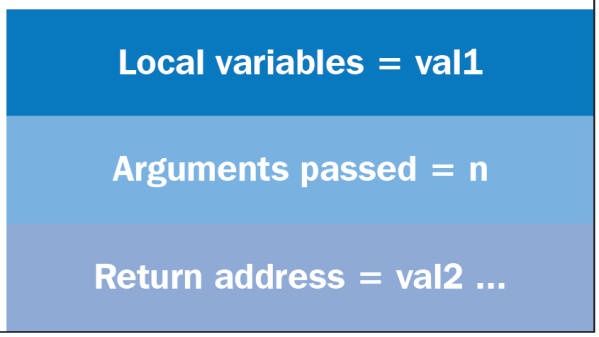

LiClipse Workspace - DFS/StackDFS.py - LiClipse (UNREGISTERED)
File Edit Refactoring Source Navigate Search Project Pydev Run Window Help
Quick Access
PyDe
DFS
d1
d2
d3
Do it yourself Code
FibonacciExample.
Node.py
NodeTest.py
RecursiveDFS.py
Stack.py
StackDFS.py
State.py
StateTest.py
python (C:\Pythor
/Python27
External Libs
Forced builtins
Predefined Cor
System Libs
State
StateTest
NodeTest
Stack
StackDFS
1 '''
2 @author: Devangini Patel
3 '''
4
5 from Node import Node
6 from State import State
7
8
9 def performStackDFS():
Console
<terminated> StackDFS.py [C:\Python27\python.exe]
reached goal state
0 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\.project
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\.pydevproject
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code
2 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\Node.py
2 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\Node.pyc
2 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\RecursiveDFS.py
2 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\State.py
2 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\State.pyc
2 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1
3 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d11
4 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d11\f111.txt
3 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d12
4 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d12\f121.txt
4 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d12\f122.txt
3 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\f11.txt
3 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\f12.txt
2 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d2
3 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d2\d21
4 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d2\d21\f211.txt
3 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d2\d22
2 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d3
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\FibonacciExample.py
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Node.py
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Node.pyc
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\NodeTest.py
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\RecursiveDFS.py
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Stack.py
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\StackDFS.py
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\State.py

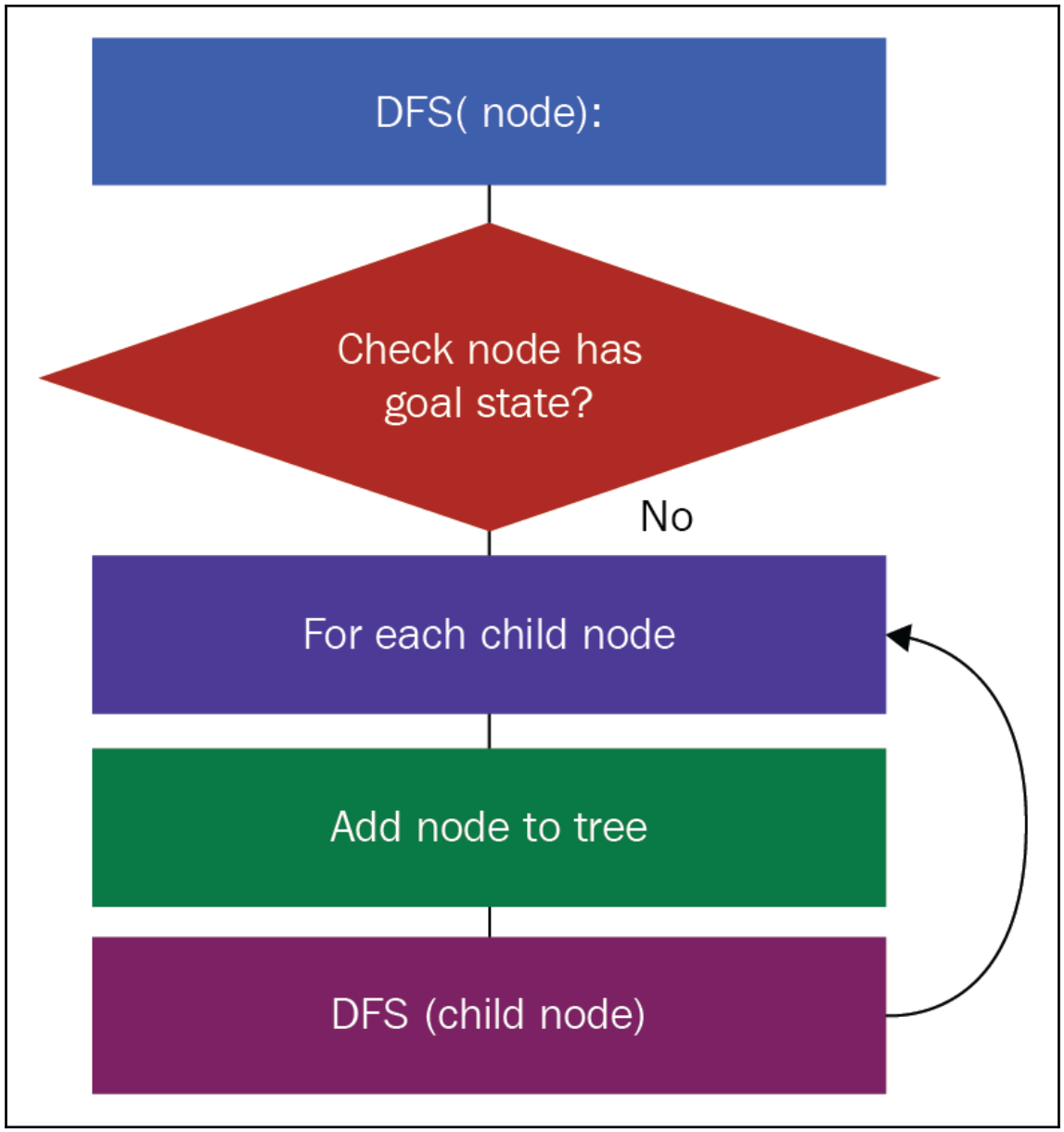

LiClipse Workspace - DFS/FibonacciExample.py - LiClipse (UNREGISTERED)
File Edit Refactoring Source Navigate Search Project Pydev Run Window Help
Quick Access
PyDe
DFS
d1
d2
d3
Do it yourself Code
FibonacciExample.
Node.py
NodeTest.py
RecursiveDFS.py
Stack.py
StackDFS.py
State.py
StateTest.py
python (C:\Pythor
/Python27
External Libs
Forced builtins
Predefined Corr
System Libs
State
StateTest
NodeTest
Stack
StackDFS
FibonacciExample
'''
@author: Devangini Patel
'''
def fibonacci(n):
if n <= 2:
return 1
else:
return fibonacci(n-1) + fibonacci(n-2)
print "fibonacci(5)", fibonacci(5)
type filter t
fibonacc
Console
<terminated> FibonacciExample.py [C:\Python27\python.exe]
fibonacci(5) 5

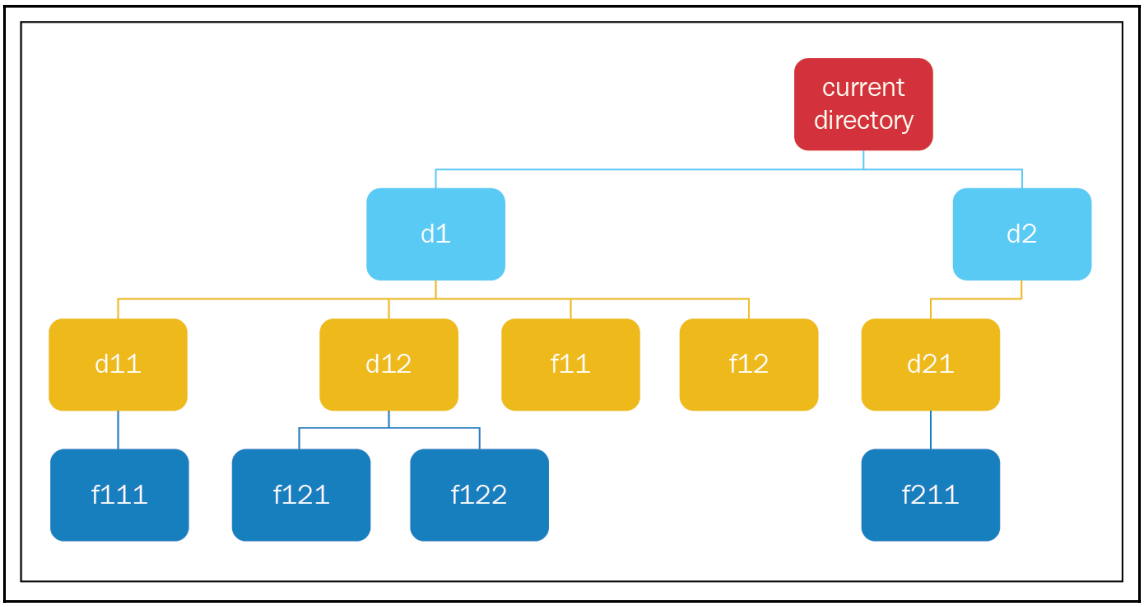

function f(n = 5){
Fibonacci Numbers
Run
if (n <= 1) /* base case */
return n;
else /* recursive caseS */
return f(n-1) + f(n-2);
var a1 =
var a2 =

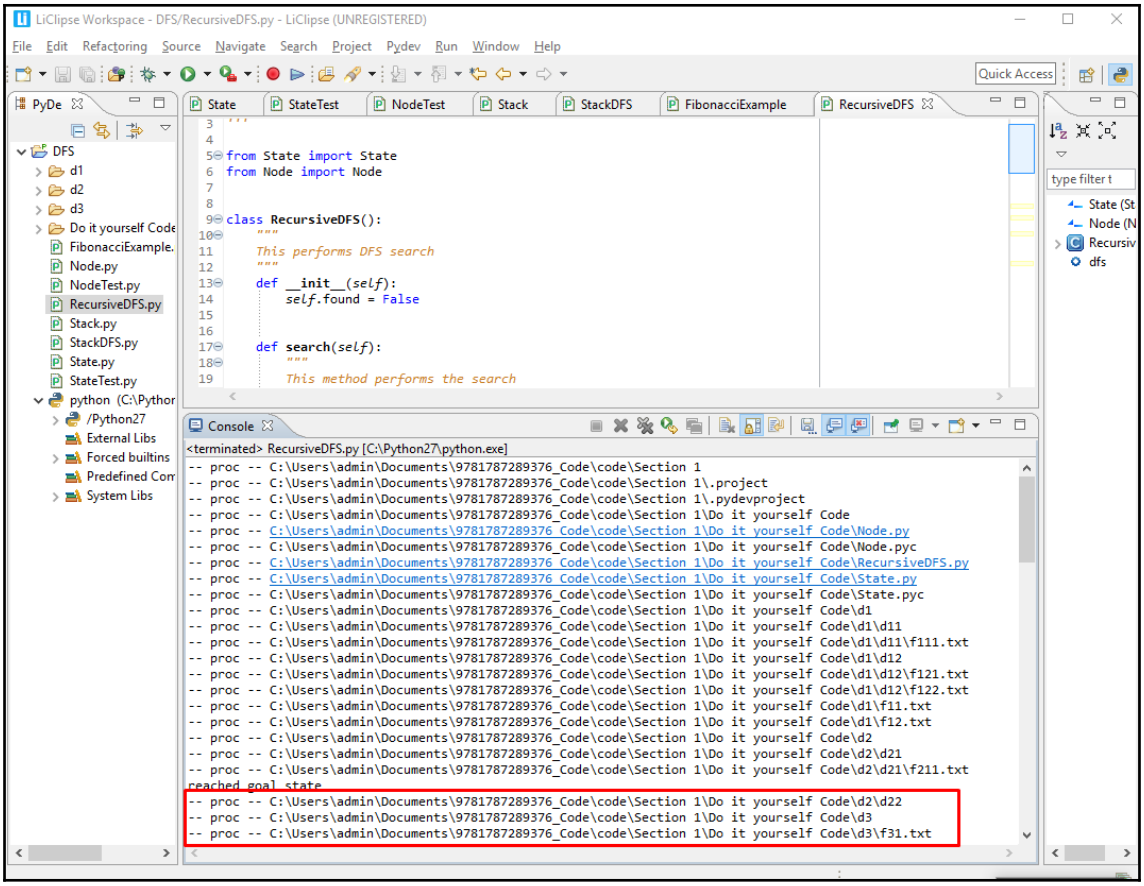
What Happens When a Function1 Calls a Function2?

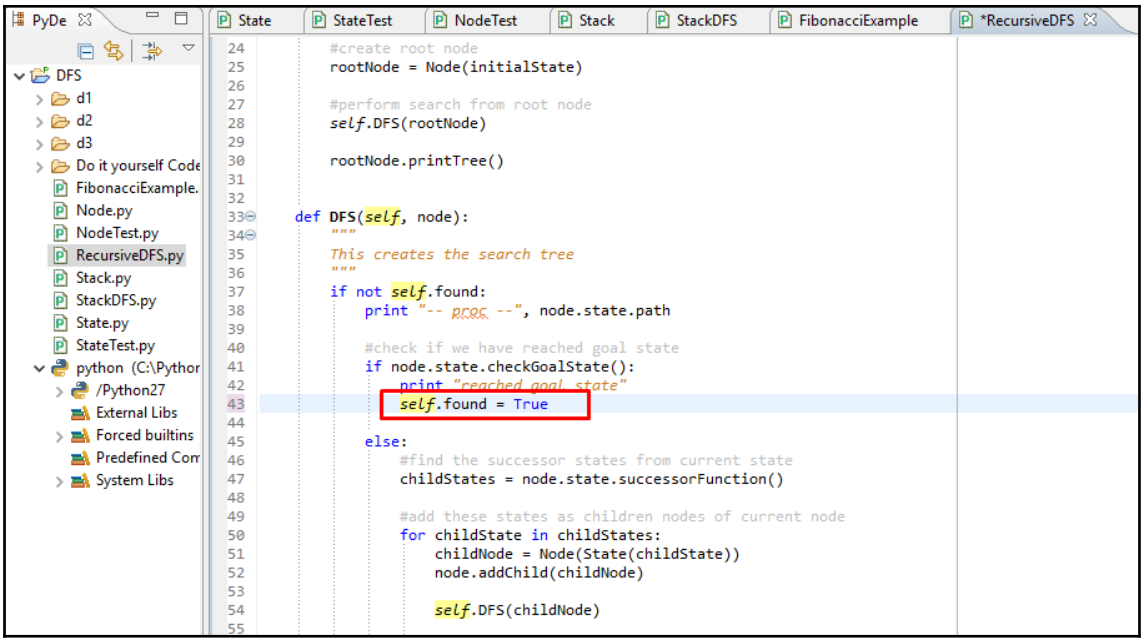
```
def fibonacci (n):
    if n <= 2:
        return 1
    else:
        val1 = fibonacci (n-1)
        val2 = fibonacci (n-2)
        val = val1 + val2
        return val
```

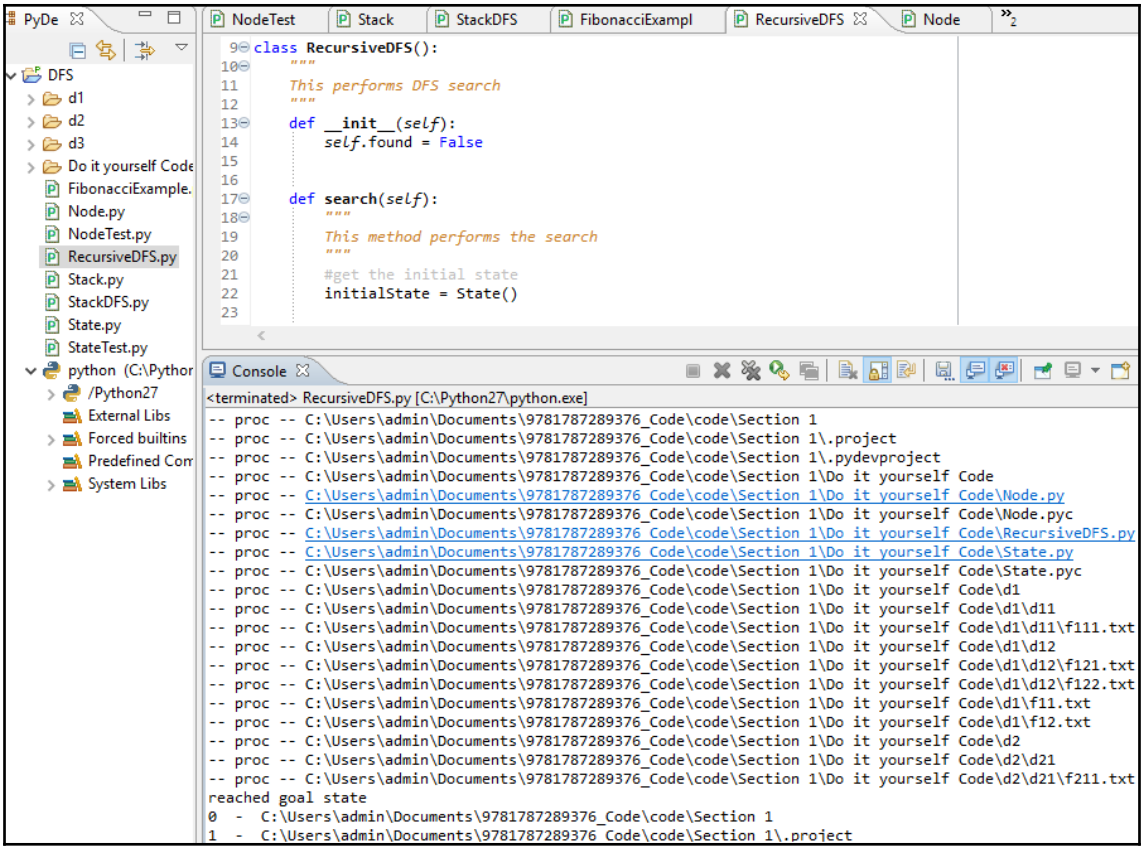



DFS(node):
Check node has goal state?
No
For each child node
Add node to tree
DFS (child node)

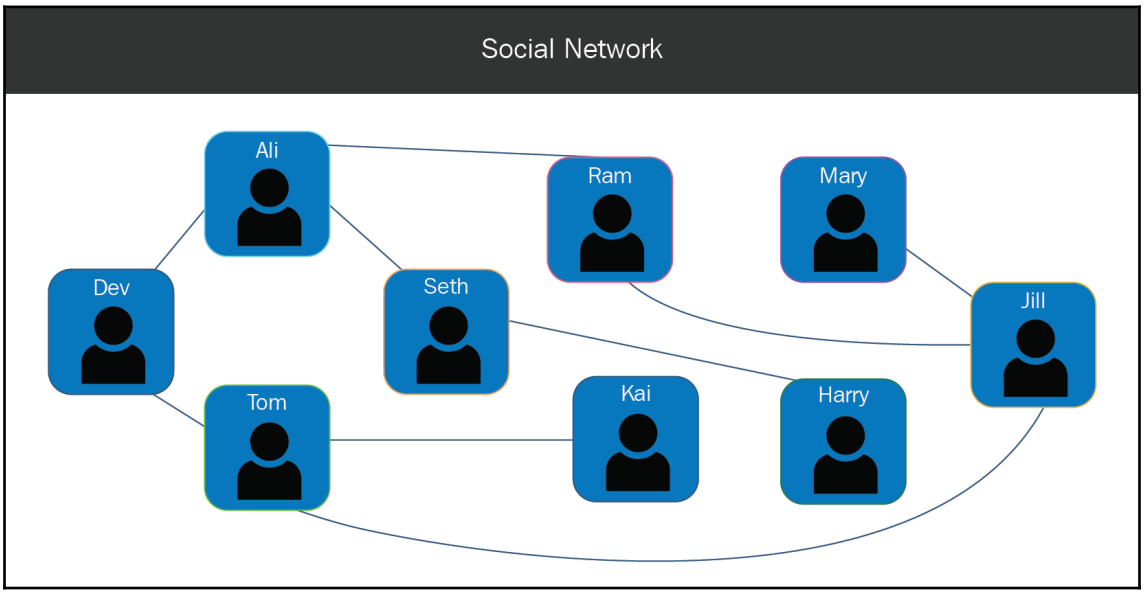

current directory
d1
d2
d11
d12
f11
f12
d21
f111
f121
f122
f211

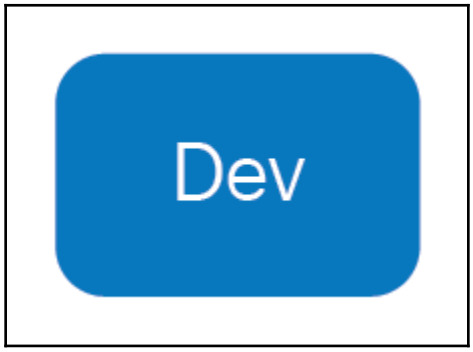

LiClipse Workspace - DFS/RecursiveDFS.py - LiClipse (UNREGISTERED)
File Edit Refactoring Source Navigate Search Project Pydev Run Window Help
Quick Access
PyDe
DFS
d1
d2
d3
Do it yourself Code
FibonacciExample.
Node.py
NodeTest.py
RecursiveDFS.py
Stack.py
StackDFS.py
State.py
StateTest.py
python (C:\Pythor
/Python27
External Libs
Forced builtins
Predefined Cor
System Libs
State
StateTest
NodeTest
Stack
StackDFS
FibonacciExample
RecursiveDFS
3 '''
4
5 from State import State
6 from Node import Node
7
8
9 class RecursiveDFS():
10 """
11 This performs DFS search
12 """
13 def __init__(self):
14 self.found = False
15
16
17 def search(self):
18 """
19 This method performs the search
Console
<terminated> RecursiveDFS.py [C:\Python27\python.exe]
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\.project
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\.pydevproject
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\Node.py
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\Node.pyc
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\RecursiveDFS.py
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\State.py
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\State.pyc
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d11
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d11\f111.txt
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d12
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d12\f121.txt
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d12\f122.txt
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\f11.txt
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\f12.txt
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d2
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d2\d21
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d2\d21\f211.txt
reached goal state
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d2\d22
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d3
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d3\f31.txt
type filter t
State (St
Node (N
Recursiv
dfs

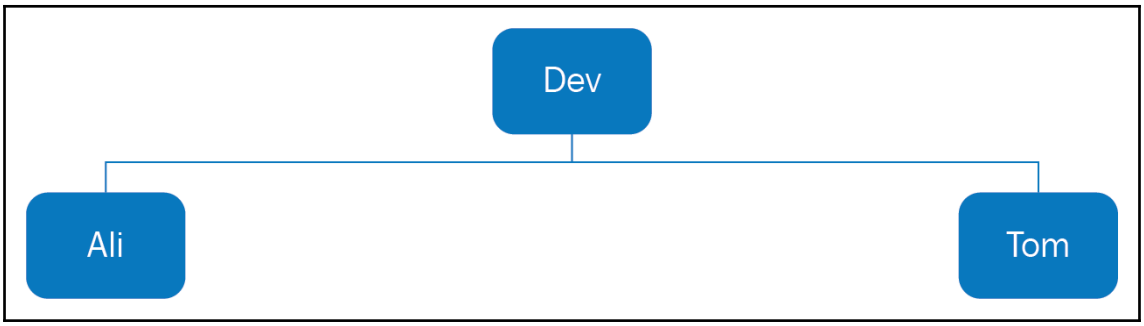

PyDe
DFS
d1
d2
d3
Do it yourself Code
FibonacciExample.
Node.py
NodeTest.py
RecursiveDFS.py
Stack.py
StackDFS.py
State.py
StateTest.py
python (C:\Pythor
/Python27
External Libs
Forced builtins
Predefined Corr
System Libs
State
StateTest
NodeTest
Stack
StackDFS
FibonacciExample
*RecursiveDFS
#create root node
rootNode = Node(initialState)
#perform search from root node
self.DFS(rootNode)
rootNode.printTree()
def DFS(self, node):
"""
This creates the search tree
"""
if not self.found:
print "-- proc --", node.state.path
#check if we have reached goal state
if node.state.checkGoalState():
print "reached goal state"
self.found = True
else:
#find the successor states from current state
childStates = node.state.successorFunction()
#add these states as children nodes of current node
for childState in childStates:
childNode = Node(State(childState))
node.addChild(childNode)
self.DFS(childNode)

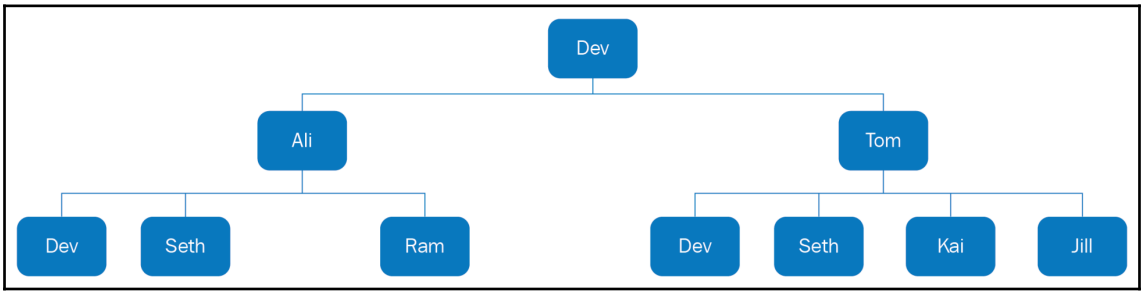

PyDe
DFS
d1
d2
d3
Do it yourself Code
FibonacciExample.
Node.py
NodeTest.py
RecursiveDFS.py
Stack.py
StackDFS.py
State.py
StateTest.py
python (C:\Pythor
/Python27
External Libs
Forced builtins
Predefined Con
System Libs
NodeTest
Stack
StackDFS
FibonacciExampl
RecursiveDFS
Node
class RecursiveDFS():
"""
This performs DFS search
"""
def __init__(self):
self.found = False
def search(self):
"""
This method performs the search
"""
#get the initial state
initialState = State()
Console
<terminated> RecursiveDFS.py [C:\Python27\python.exe]
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\.project
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\.pydevproject
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\Node.py
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\Node.pyc
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\RecursiveDFS.py
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\State.py
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\State.pyc
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d11
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d11\f111.txt
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d12
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d12\f121.txt
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\d12\f122.txt
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\f11.txt
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d1\f12.txt
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d2
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d2\d21
-- proc -- C:\Users\admin\Documents\9781787289376_Code\code\Section 1\Do it yourself Code\d2\d21\f211.txt
reached goal state
0 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1
1 - C:\Users\admin\Documents\9781787289376_Code\code\Section 1\.project

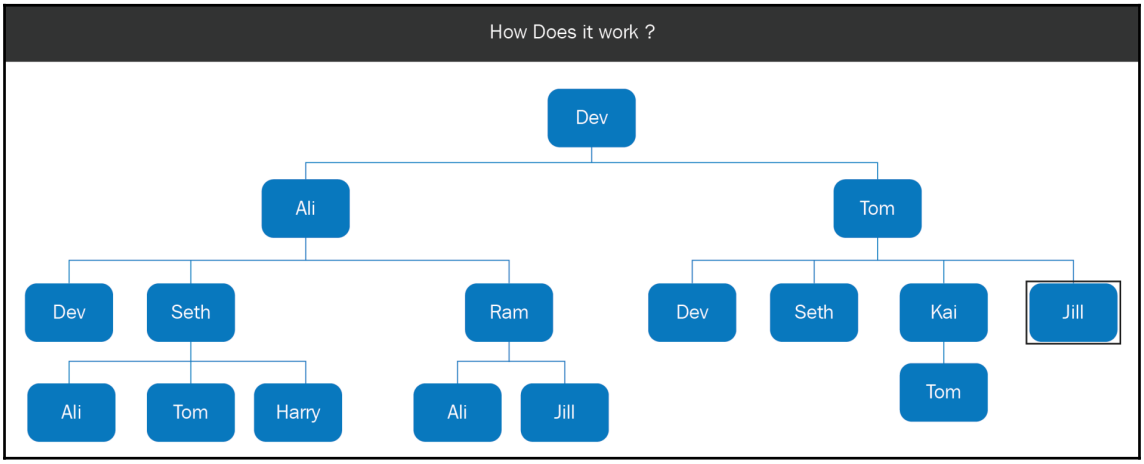
Chapter 2: Understanding the Breadth-First Search Algorithm



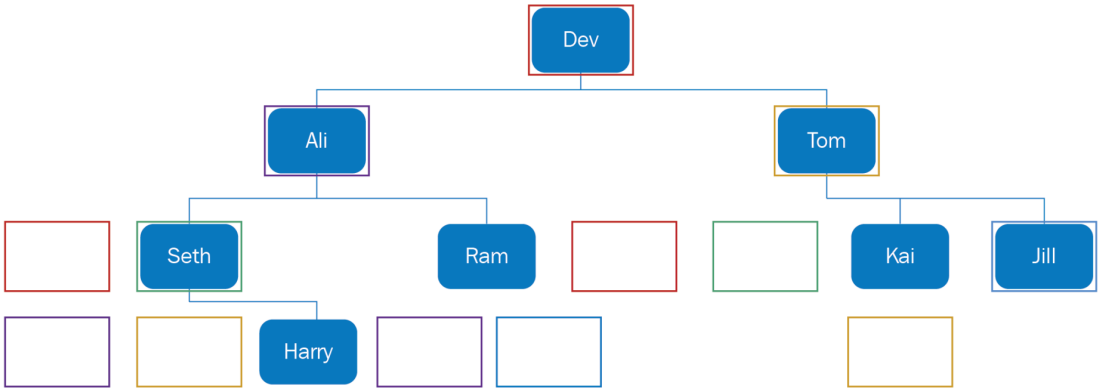



Dev
Ali
Tom

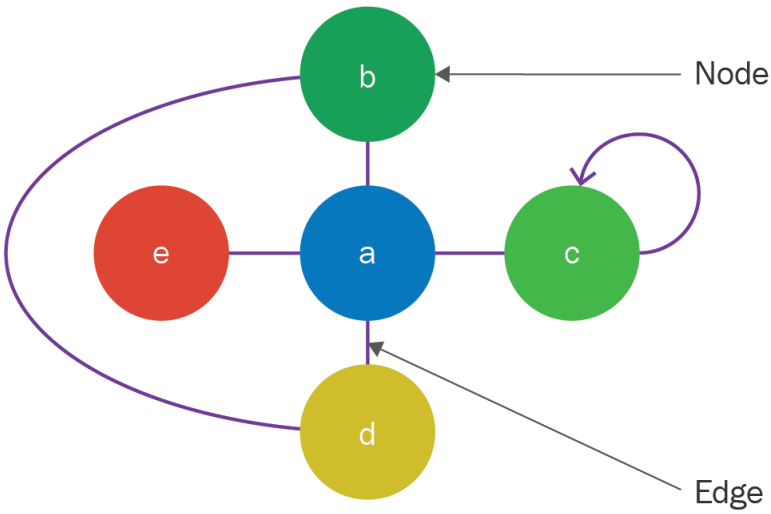

Dev
Ali
Tom
Dev
Seth
Ram
Dev
Seth
Kai
Jill


How Does it work ?
Dev
Ali
Tom
Dev
Seth
Ram
Dev
Seth
Kai
Jill
Ali
Tom
Harry
Ali
Jill
Tom

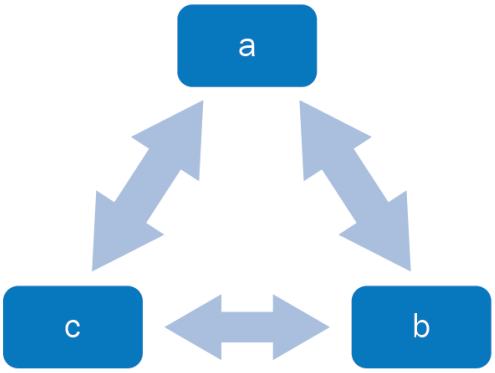
How Does it work ?



Graph



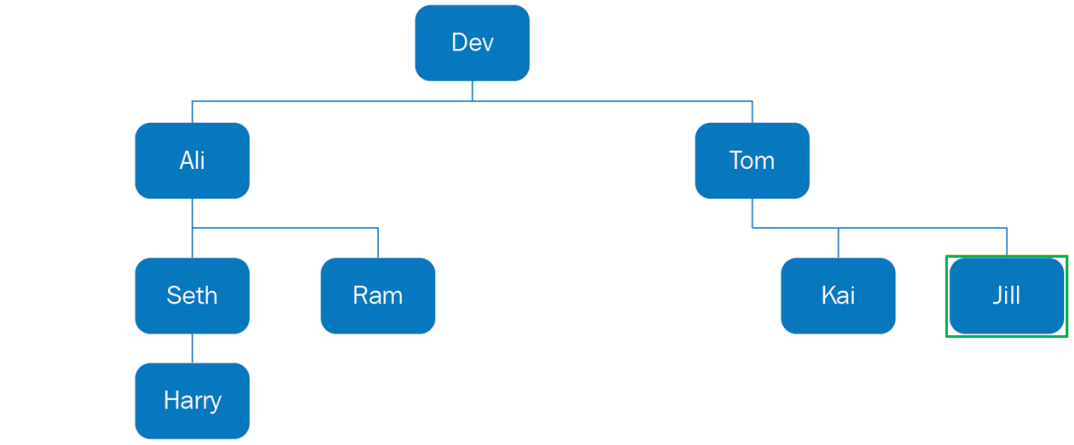
Graph in Python



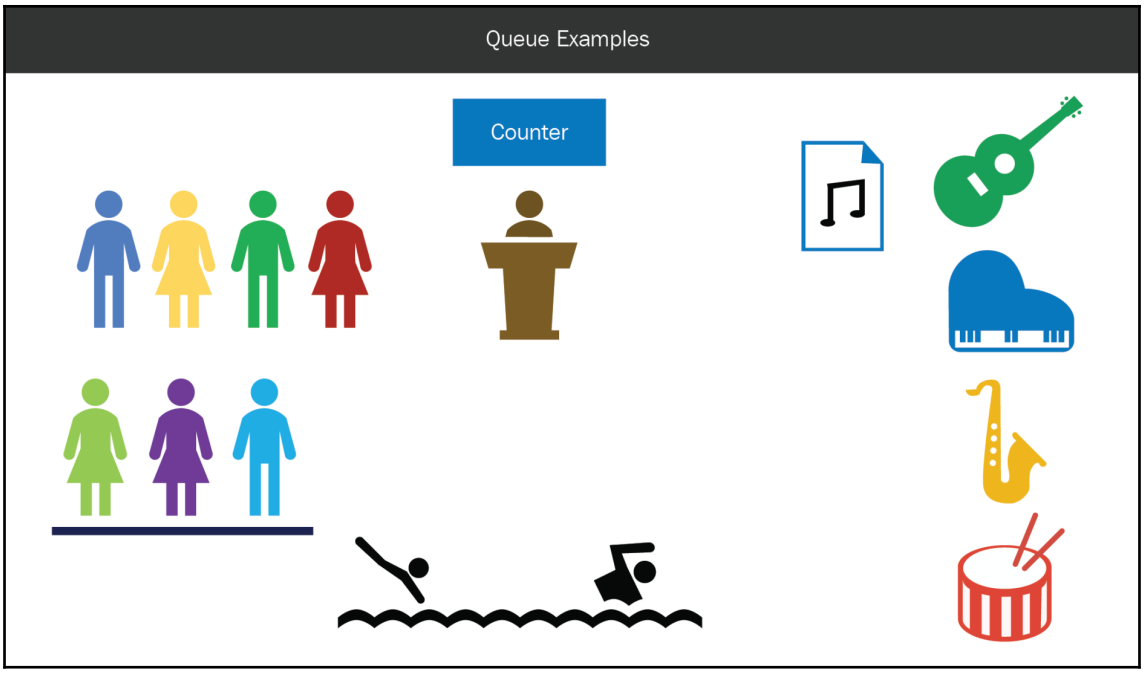
dictionary

node { a → b, c
b → c, a
c → a, b } connections

Using Graph as a Tree



visited: Dev | Ali | Seth | Tom | Ram | Harry | Kai | Jill


Queue Examples
Counter

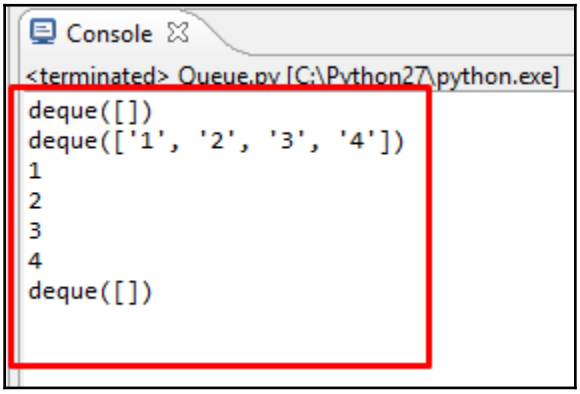
Queue Operations

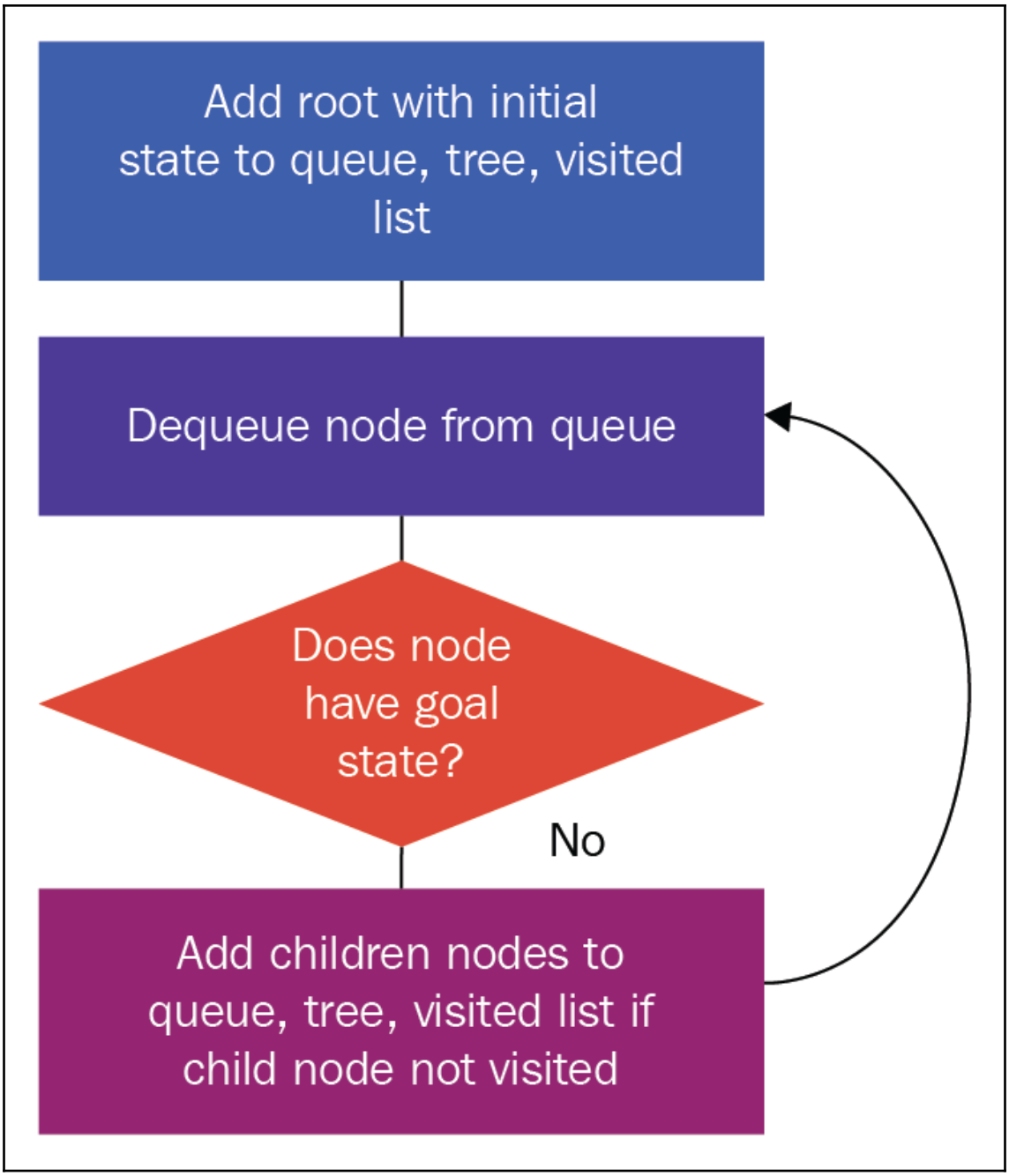
Counter

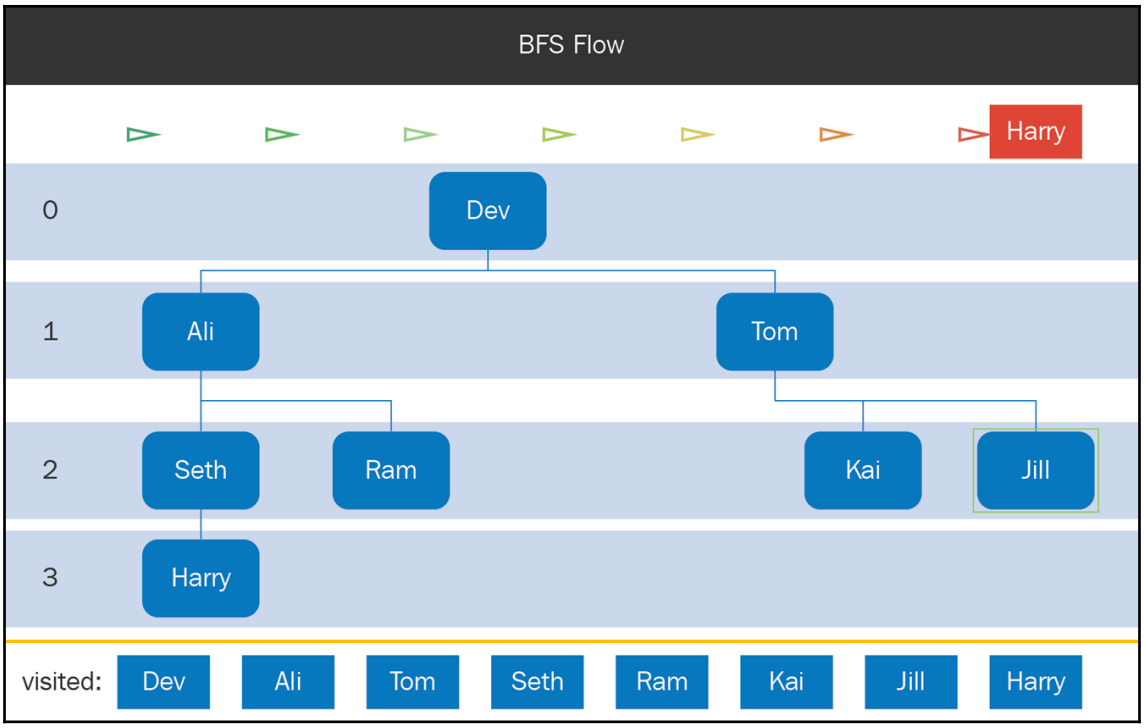
+ Enqueue
  At back

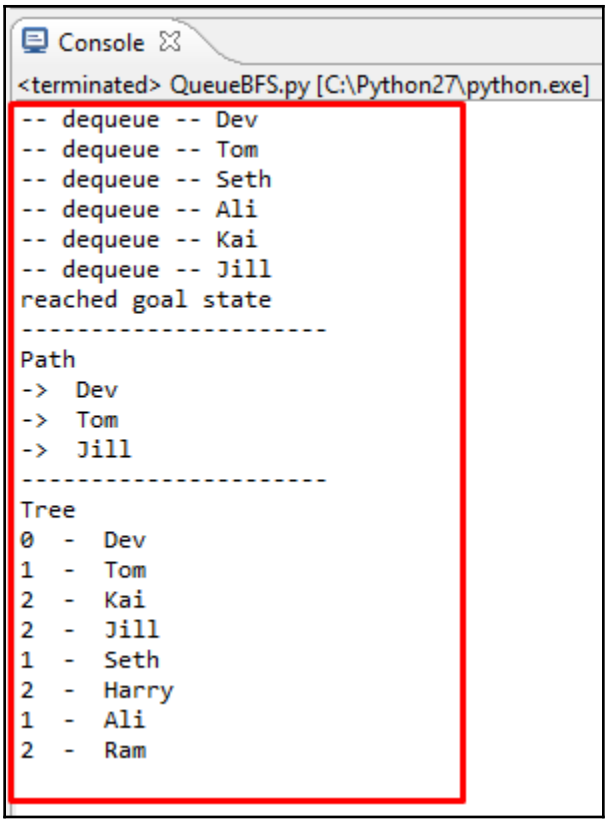
FIFO (First in first Out)

- Dequeue
  At Front



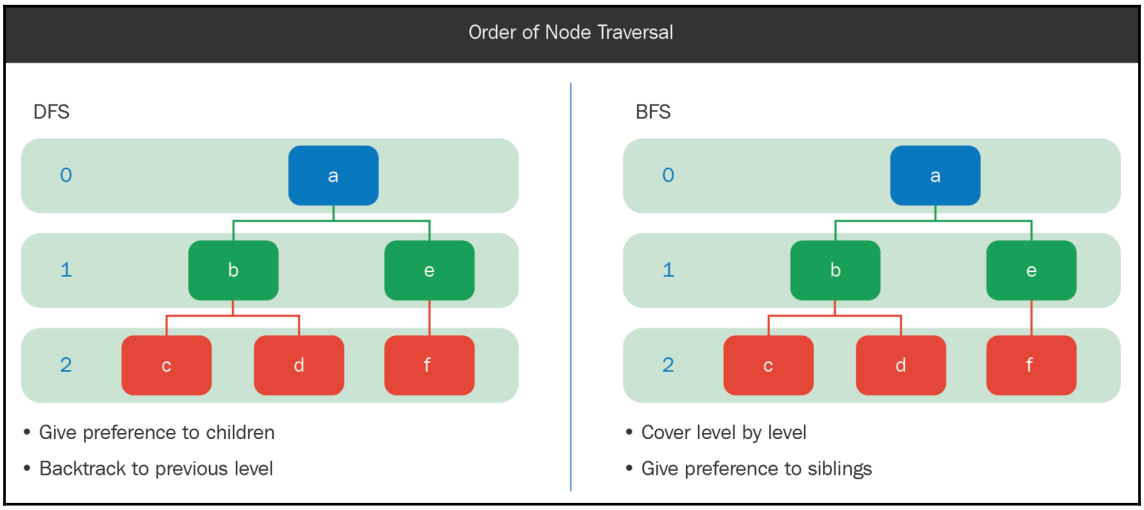

Add root with initial state to queue, tree, visited list
Dequeue node from queue
Does node have goal state?
No
Add children nodes to queue, tree, visited list if child node not visited


BFS Flow
Harry
0
Dev
1
Ali
Tom
2
Seth
Ram
Kai
Jill
3
Harry
visited:
Dev
Ali
Tom
Seth
Ram
Kai
Jill
Harry


Console
<terminated> QueueBFS.py [C:\Python27\python.exe]
-- dequeue -- Dev
-- dequeue -- Tom
-- dequeue -- Seth
-- dequeue -- Ali
-- dequeue -- Kai
-- dequeue -- Jill
reached goal state

Path
-> Dev
-> Tom
-> Jill

Tree
0 - Dev
1 - Tom
2 - Kai
2 - Jill
1 - Seth
2 - Harry
1 - Ali
2 - Ram


Order of Node Traversal
DFS
0
a
1
b
e
2
c
d
f
• Give preference to children
• Backtrack to previous level
BFS
0
a
1
b
e
2
c
d
f
• Cover level by level
• Give preference to siblings

Data Structure

DFS

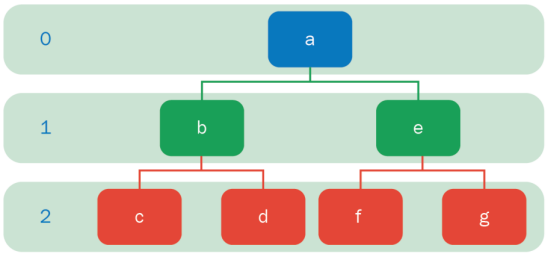
- Stack

BFS

- Queue

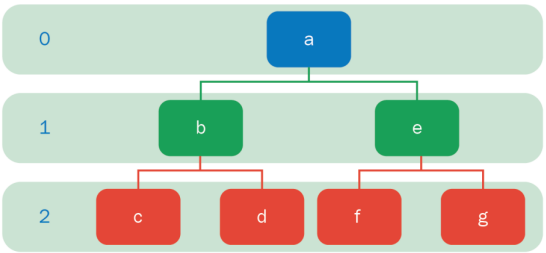
Memory Used

DFS



- DFS(c), implicit stack = [(e), (c,d)]
- O(d), d = depth

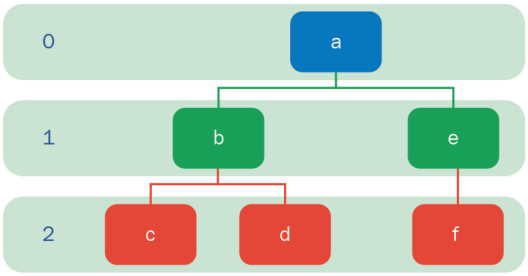
BFS



- BFS(c), queue = [c,d,f,g]
- O(b^d), b = branching factor

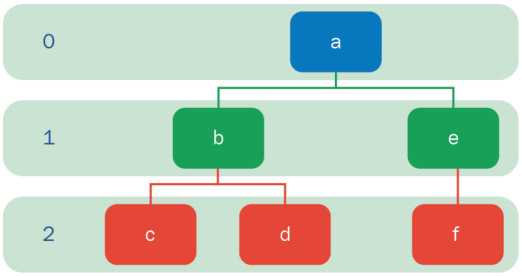
Optimal Solution

DFS



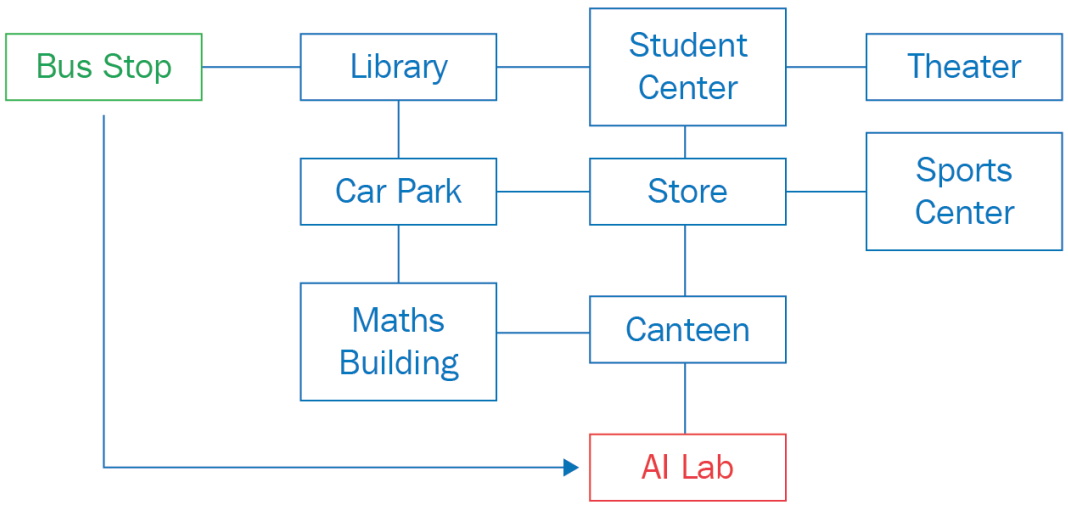
- DFS finds suboptimal solution d before optimal solution e

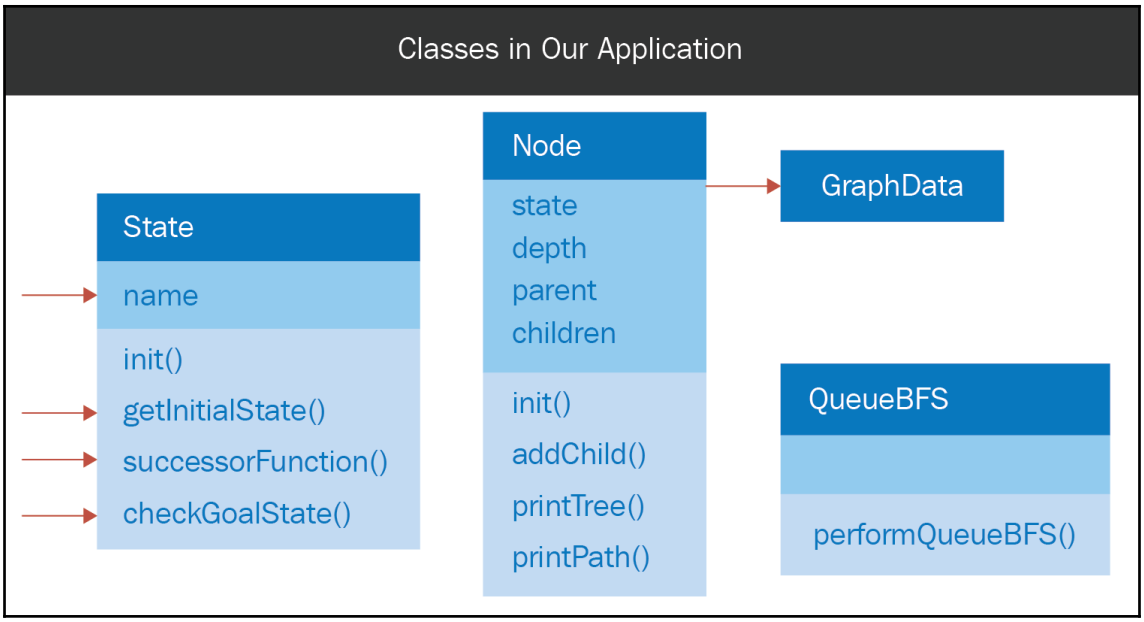
BFS

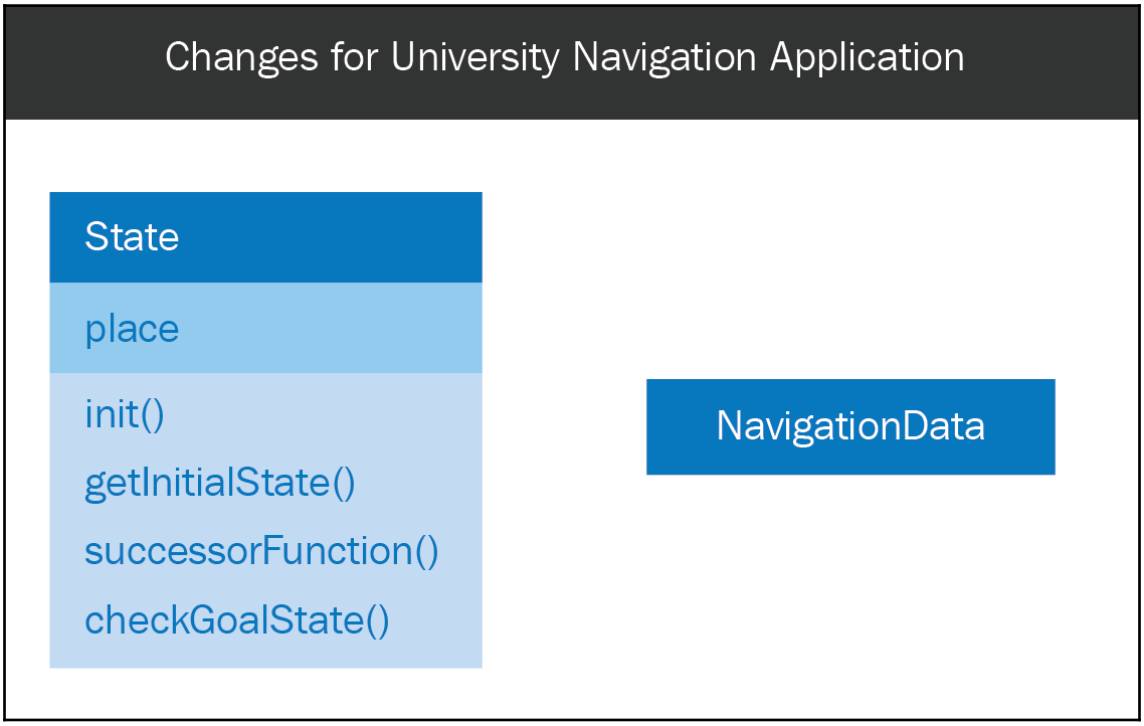


- BFS finds optimal solution e before d

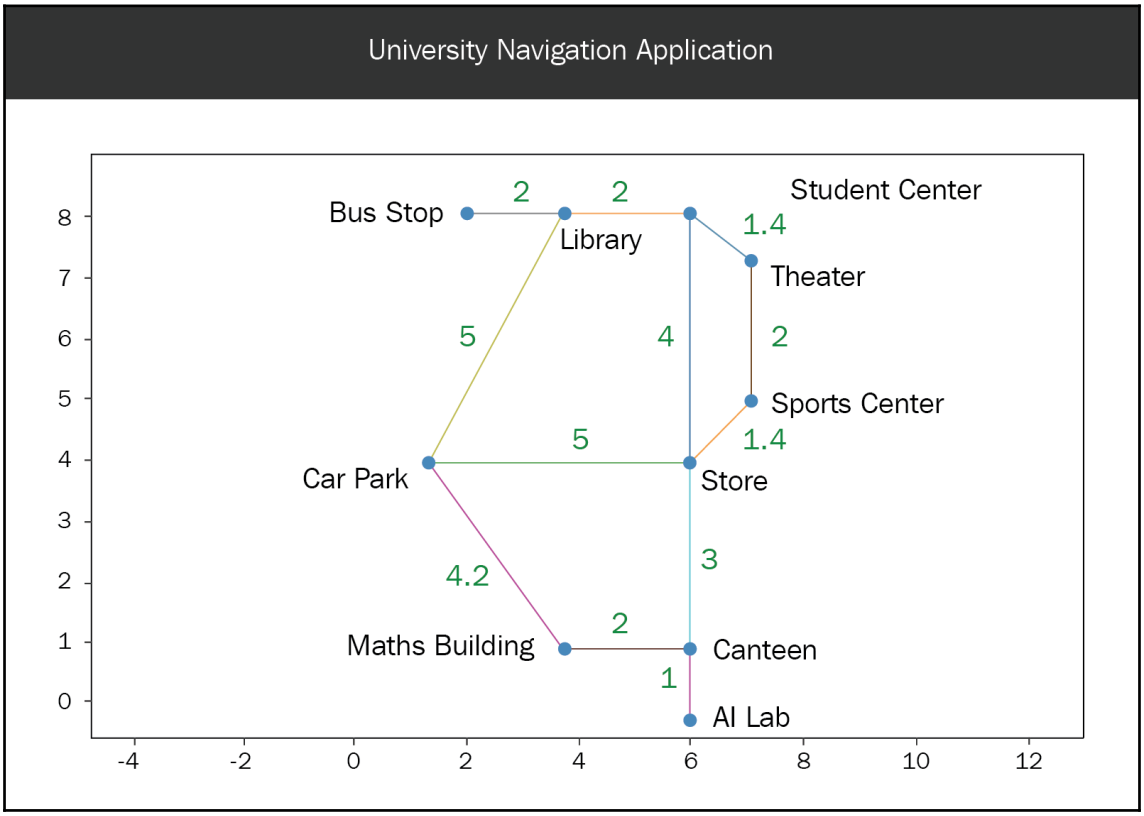
University Navigation Application

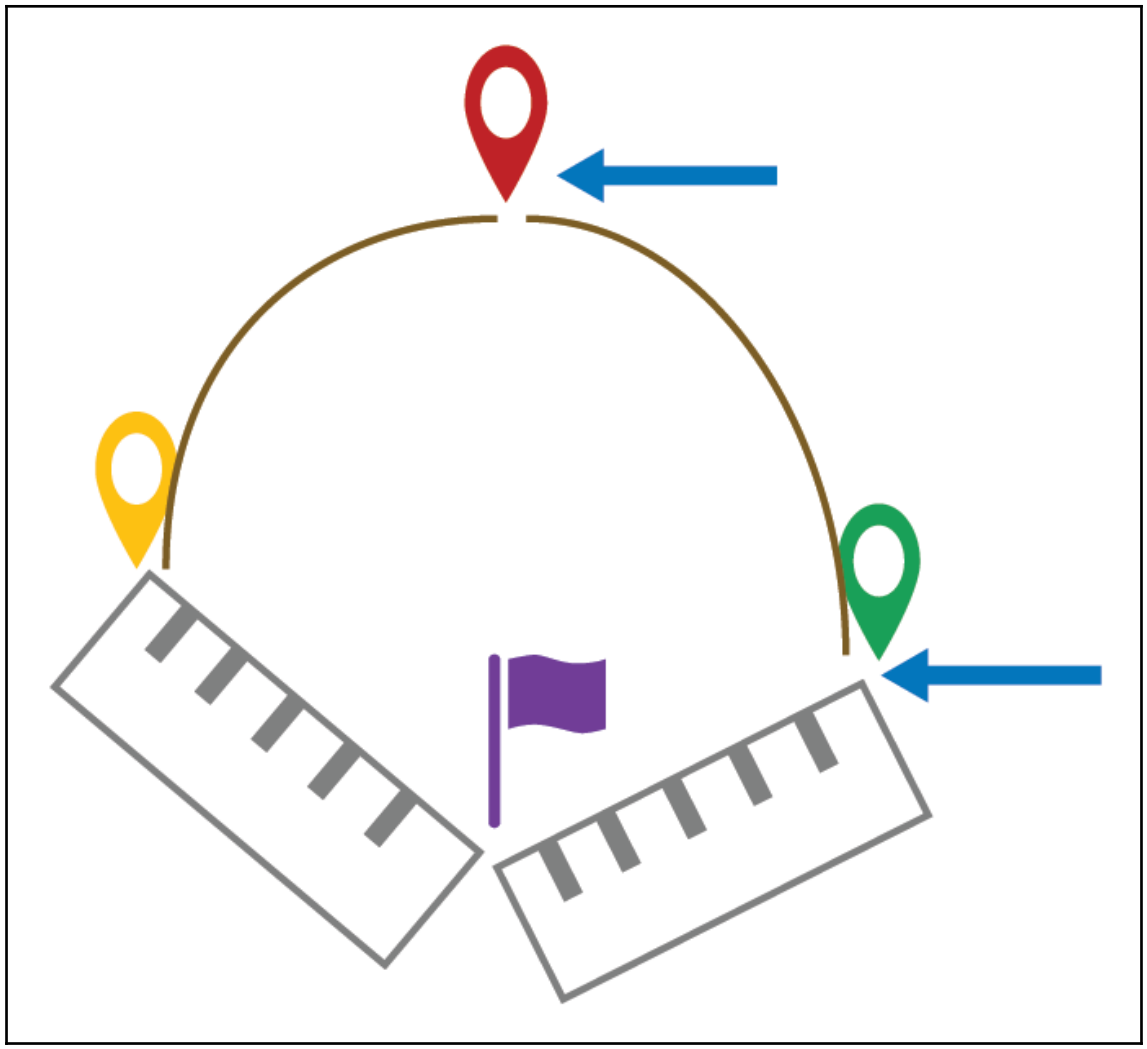


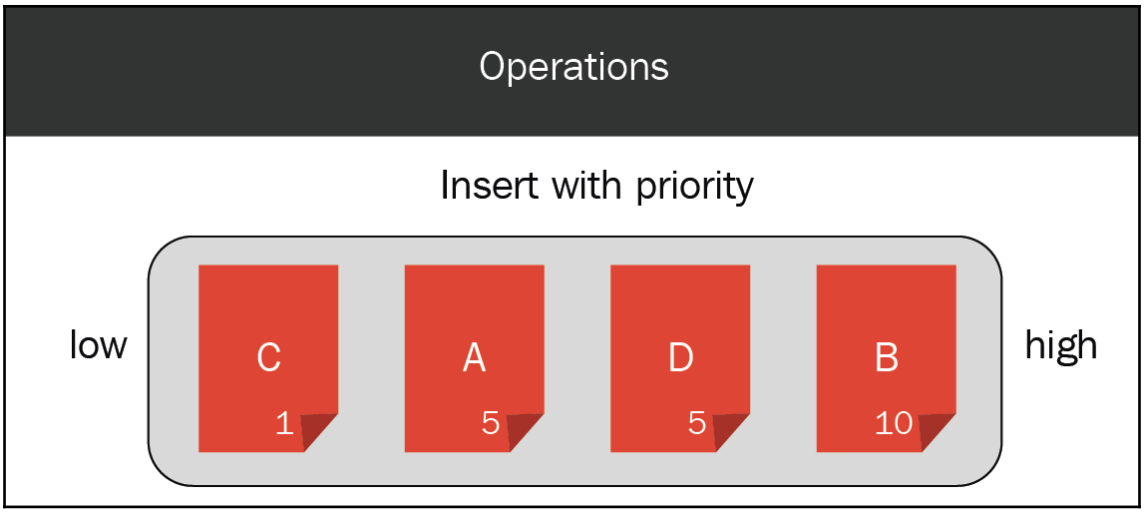

Classes in Our Application
State
name
init()
getInitialState()
successorFunction()
checkGoalState()
Node
state
depth
parent
children
init()
addChild()
printTree()
printPath()
GraphData
QueueBFS
performQueueBFS()


Changes for University Navigation Application
State
place
init()
getInitialState()
successorFunction()
checkGoalState()
NavigationData

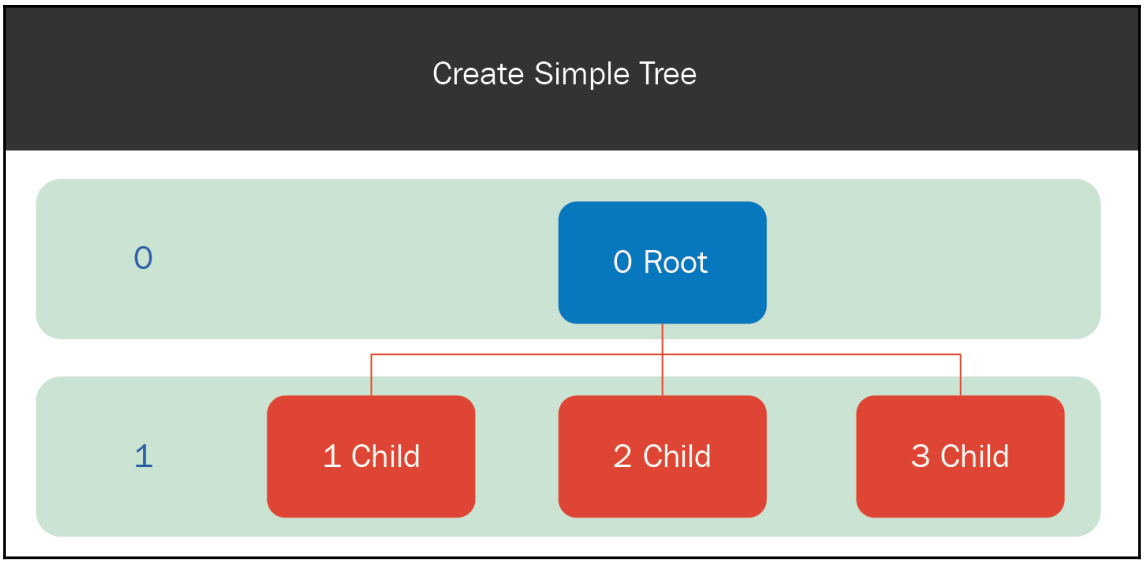
Chapter 3: Understanding the Heuristic Search Algorithm

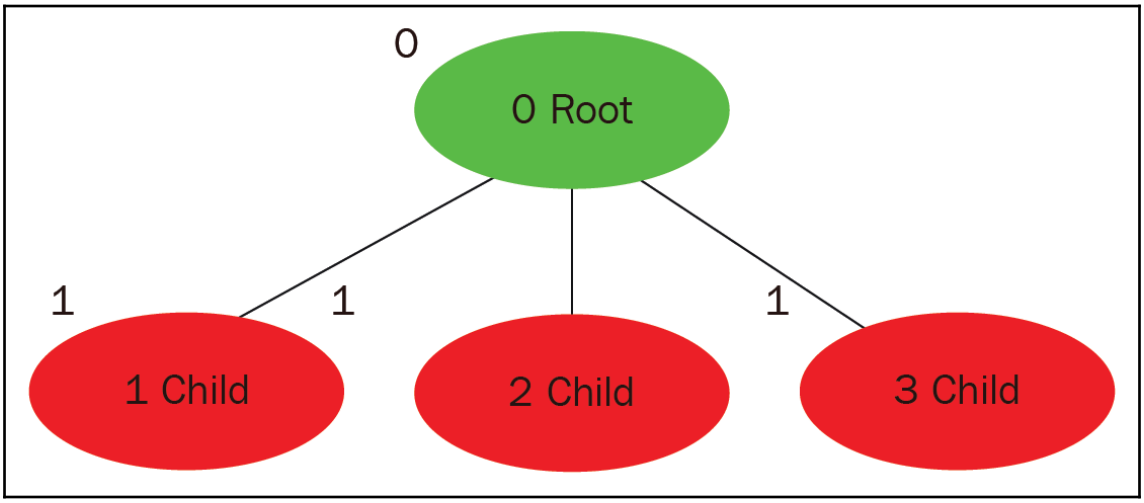


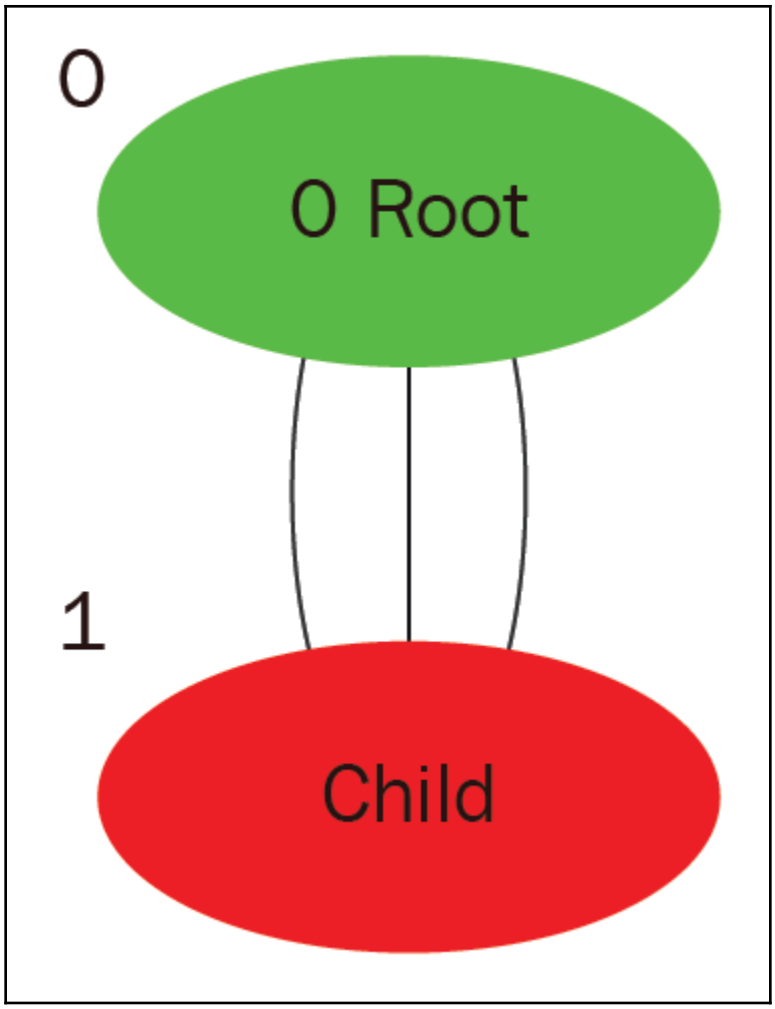


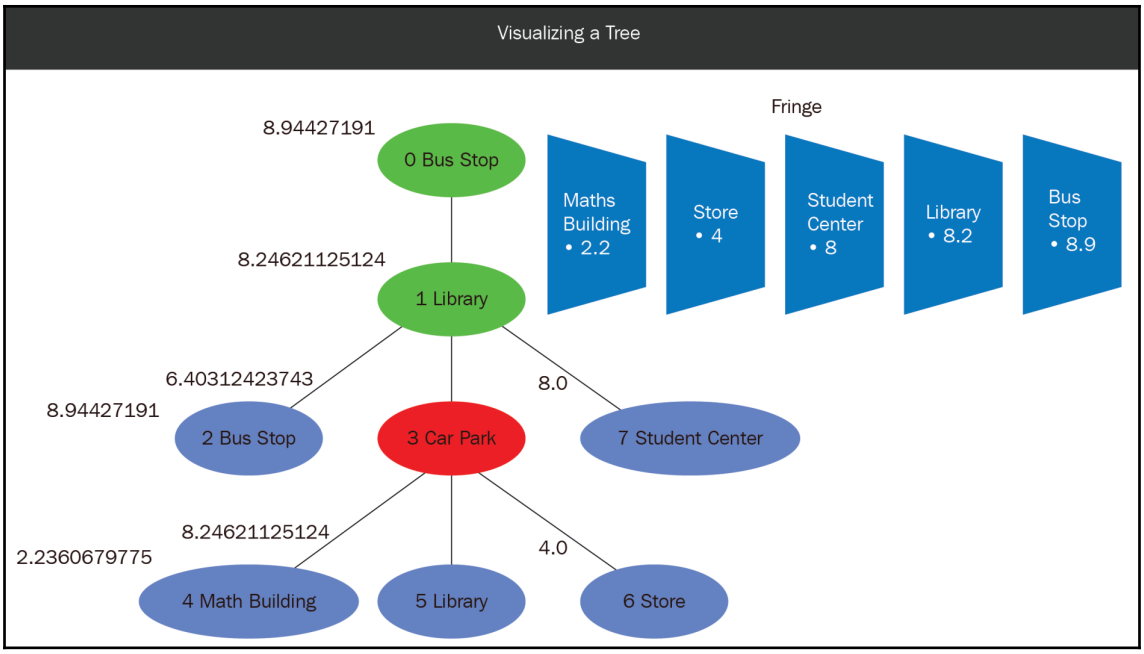


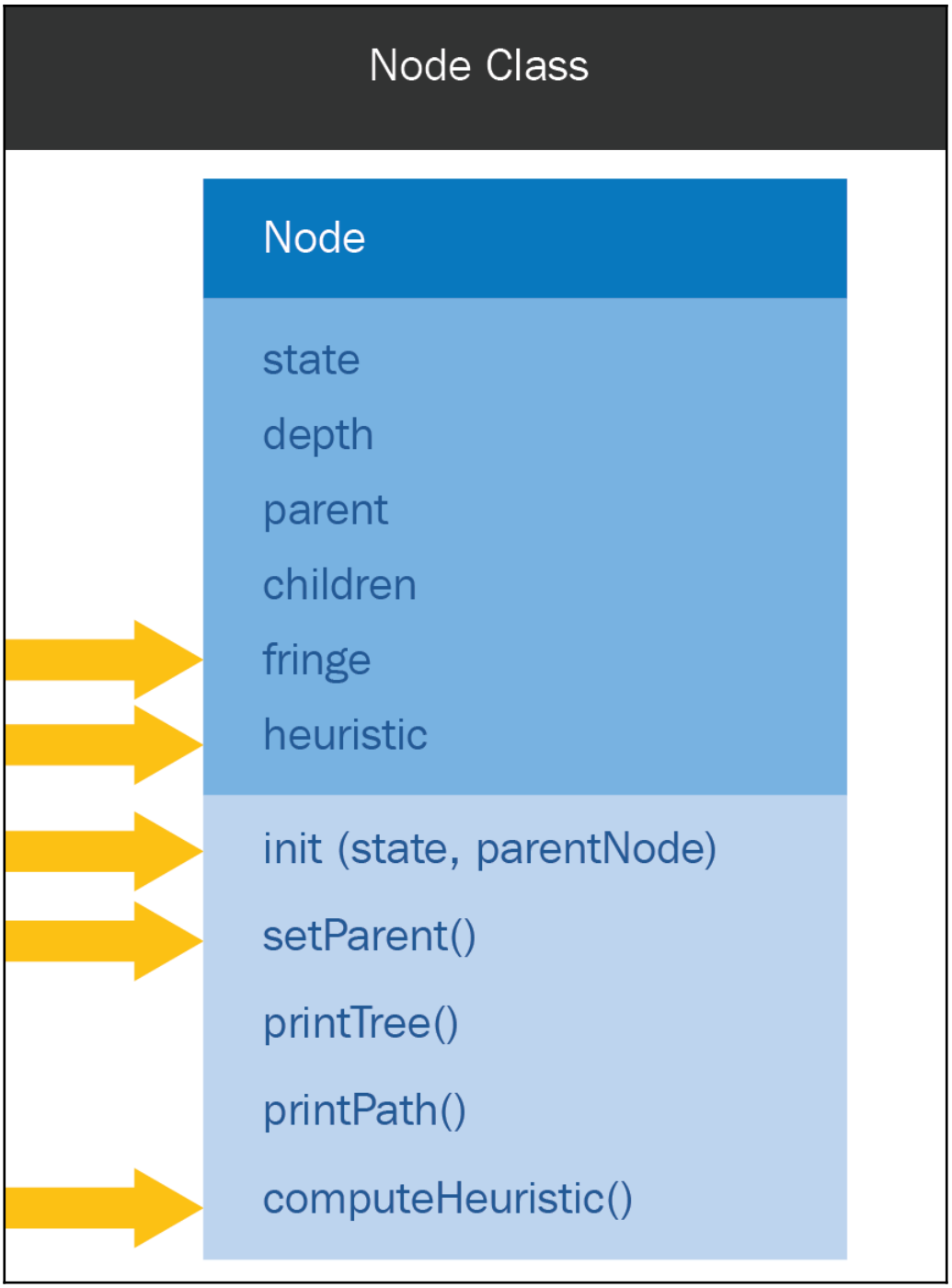

```
Console
<terminated> PriorityQueue.py [C:\Python27\python.exe]
0
4
(1, 'C')
(5, 'A')
(5, 'D')
(10, 'B')
0
```

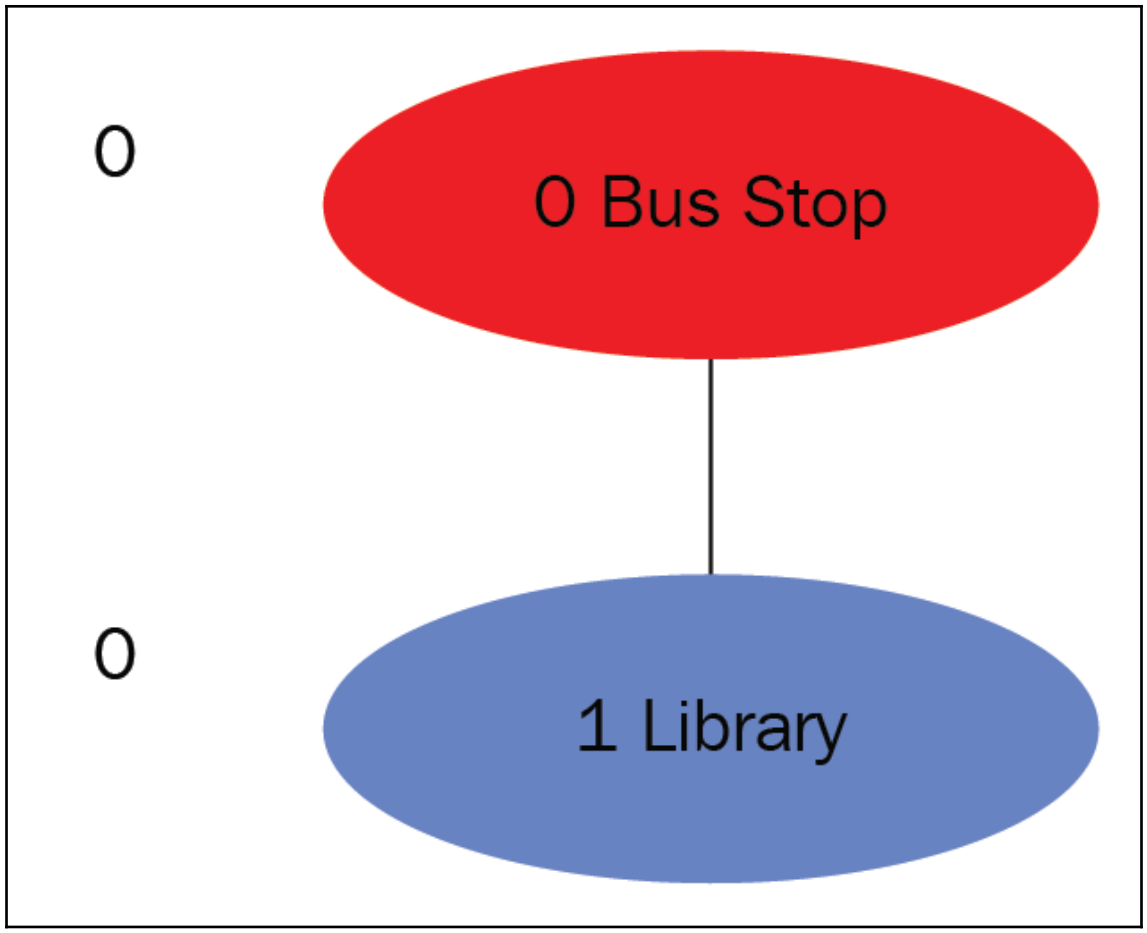

Create Simple Tree
0
0 Root
1
1 Child
2 Child
3 Child


0
0 Root
1
1
1
1 Child
2 Child
3 Child

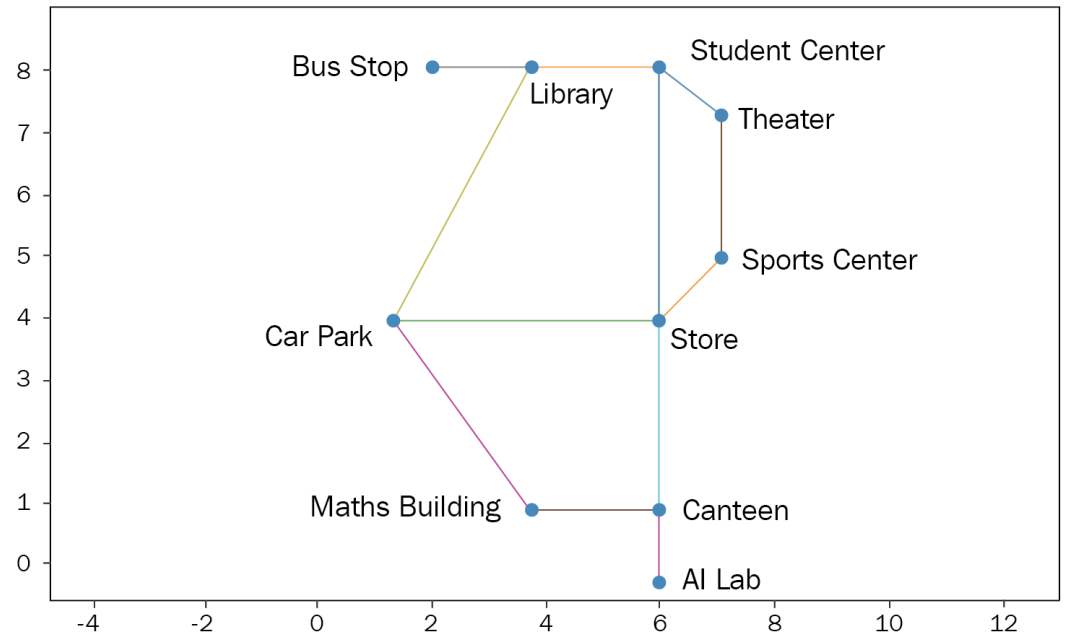

0
0 Root
1
Child

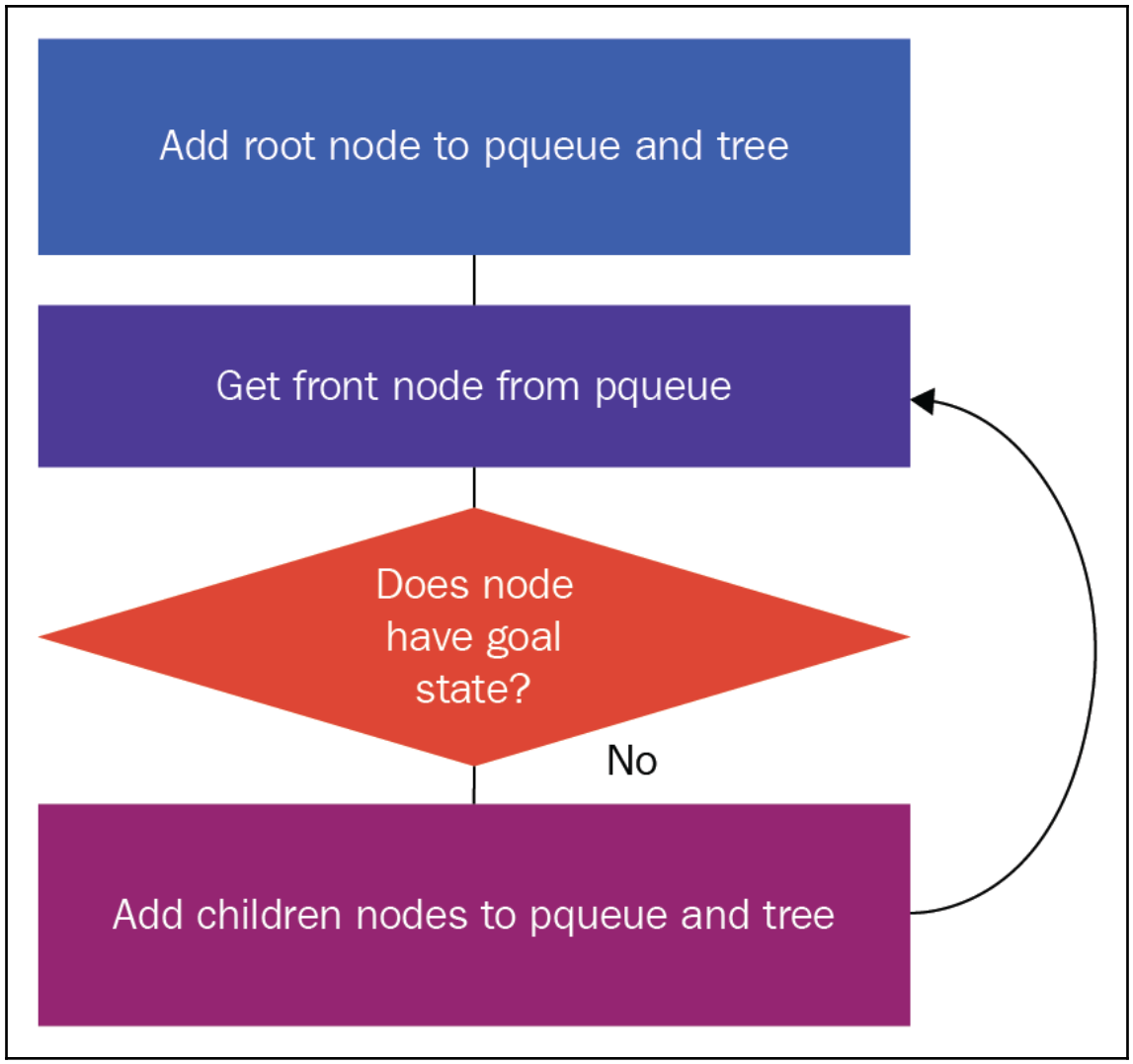

Visualizing a Tree
Fringe
8.94427191
0 Bus Stop
Maths Building • 2.2
Store • 4
Student Center • 8
Library • 8.2
Bus Stop • 8.9
8.24621125124
1 Library
6.40312423743
8.94427191
2 Bus Stop
3 Car Park
8.0
7 Student Center
8.24621125124
2.2360679775
4 Math Building
5 Library
4.0
6 Store

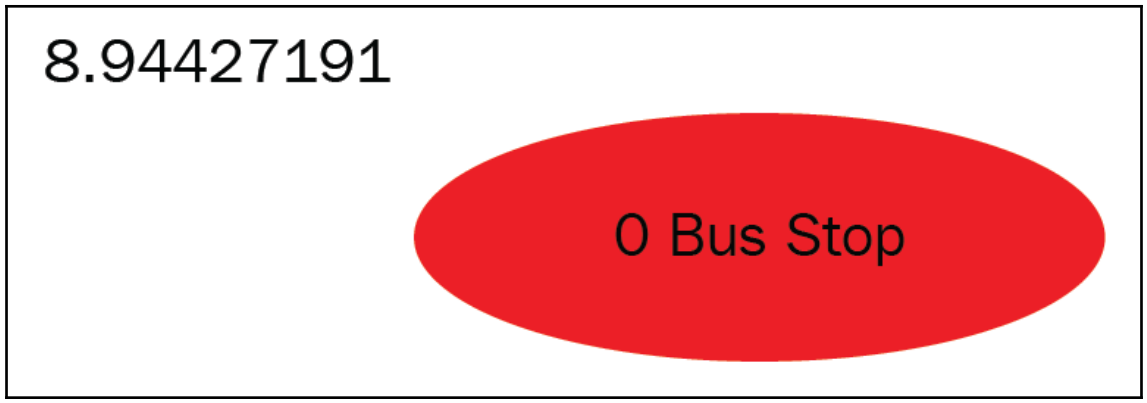

Node Class
Node
state
depth
parent
children
fringe
heuristic
init (state, parentNode)
setParent()
printTree()
printPath()
computeHeuristic()

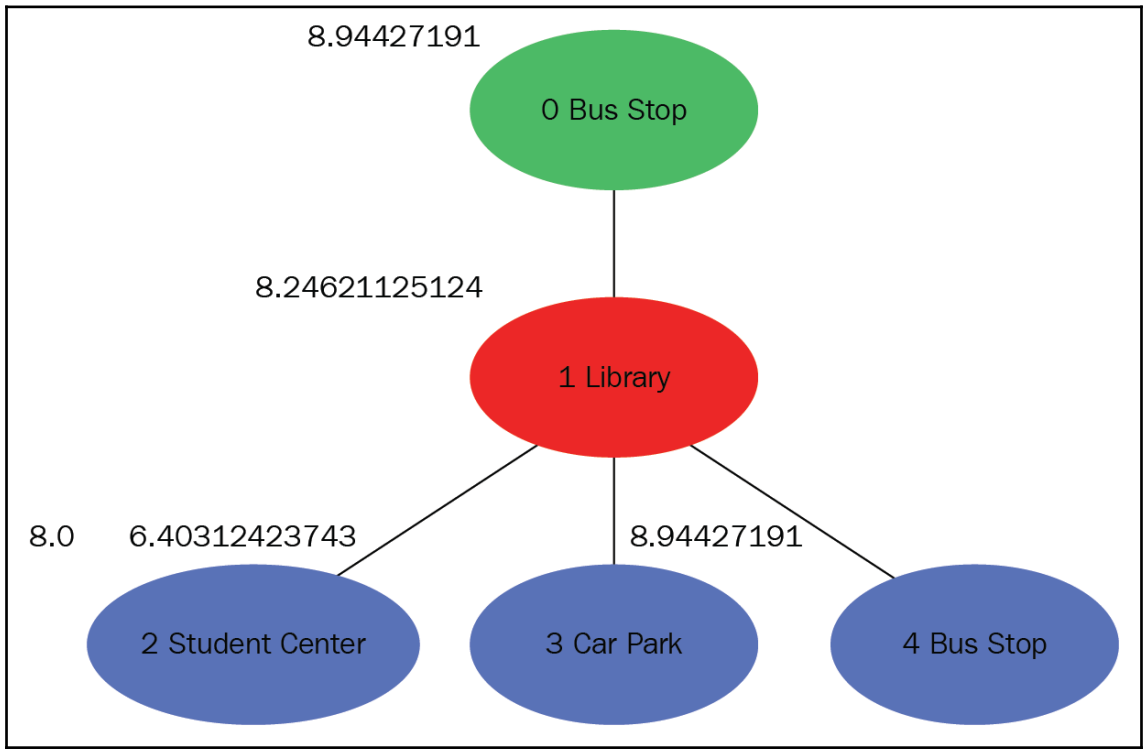

0
0 Bus Stop
0
1 Library

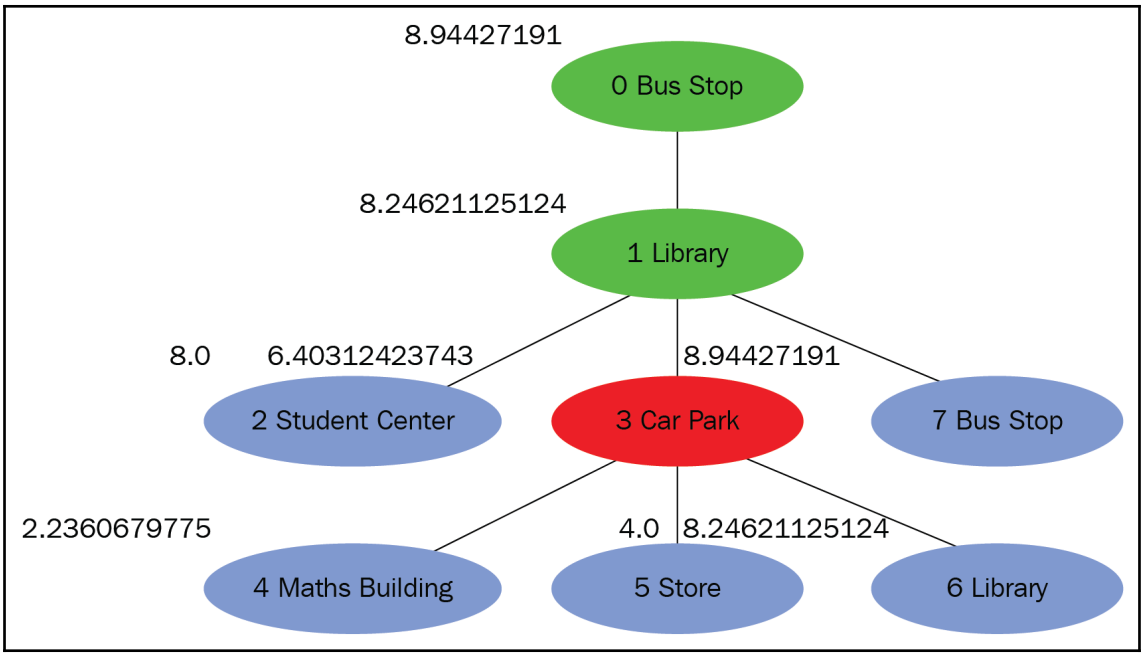
University Navigation Application

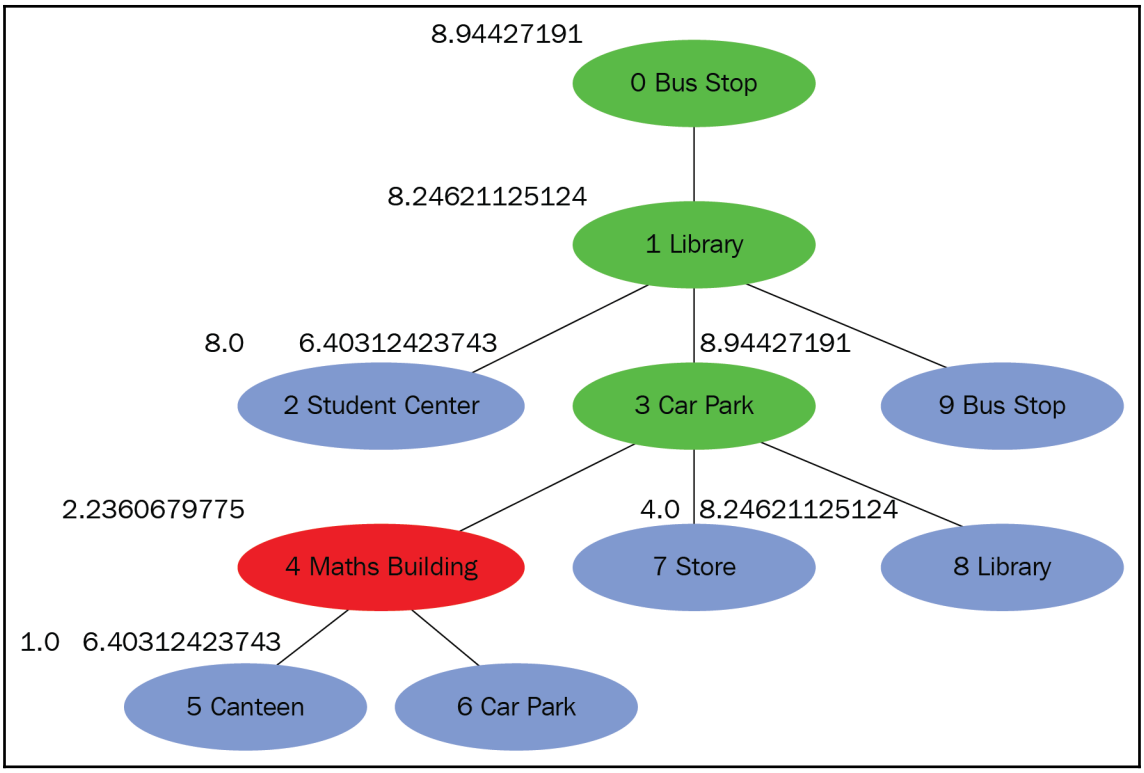


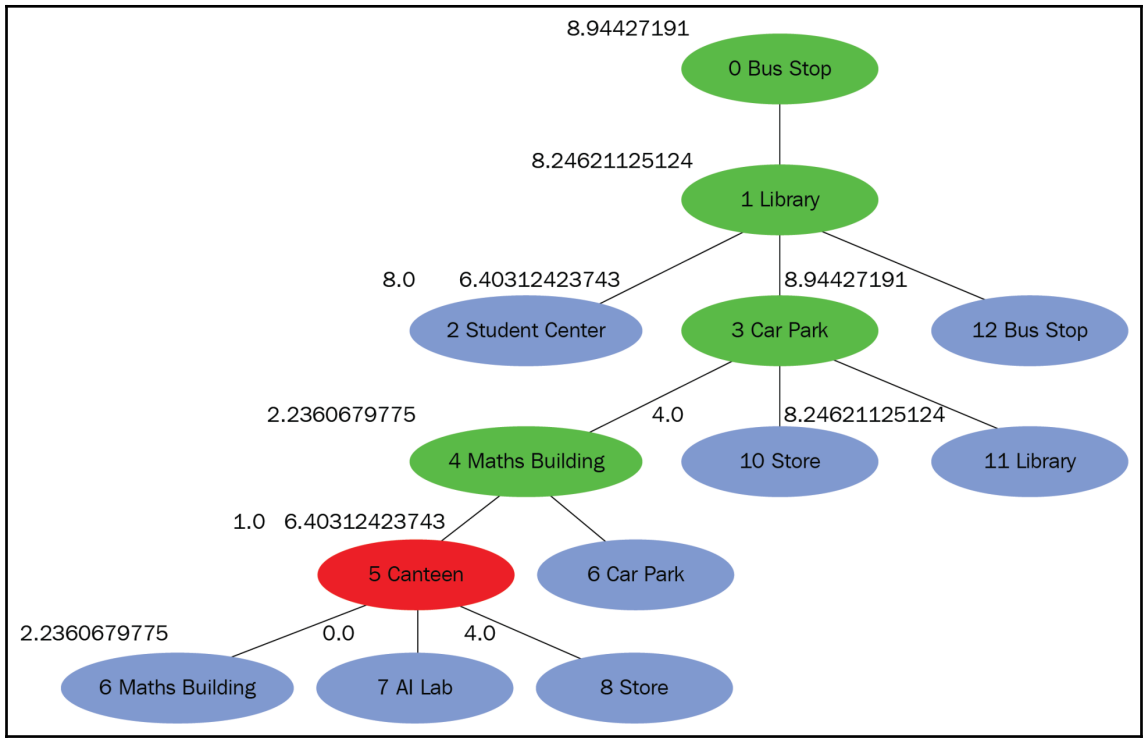

Add root node to pqueue and tree
Get front node from pqueue
Does node have goal state?
No
Add children nodes to pqueue and tree

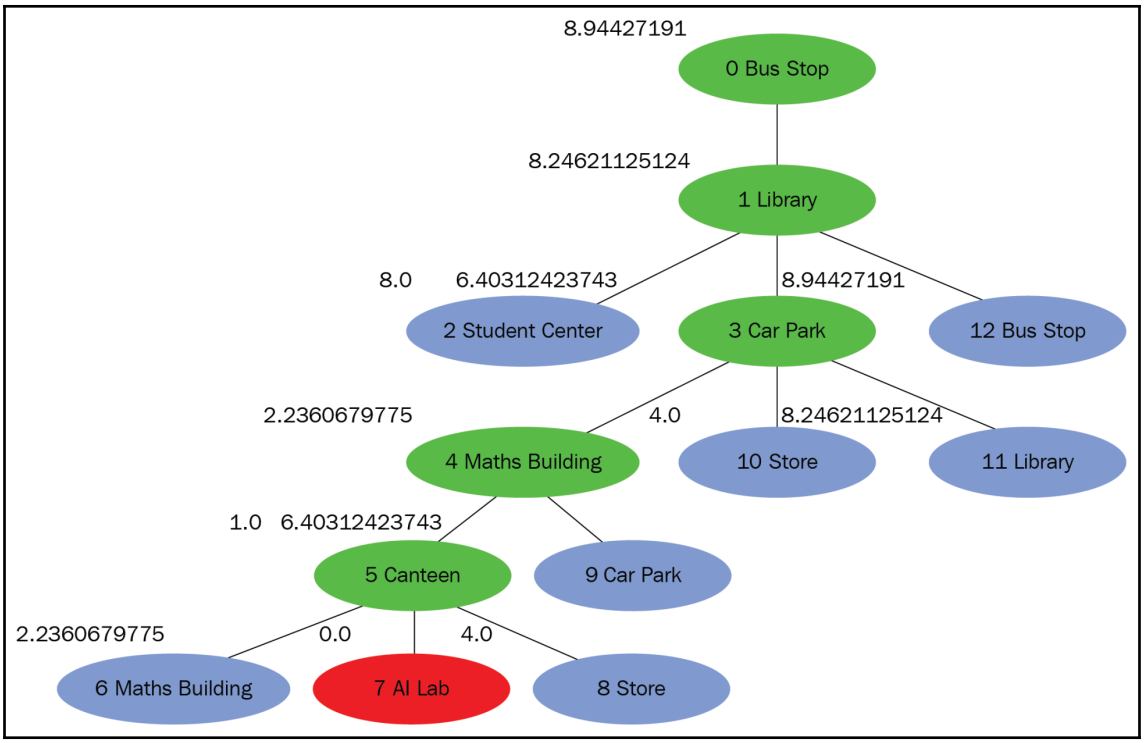

8.94427191
0 Bus Stop

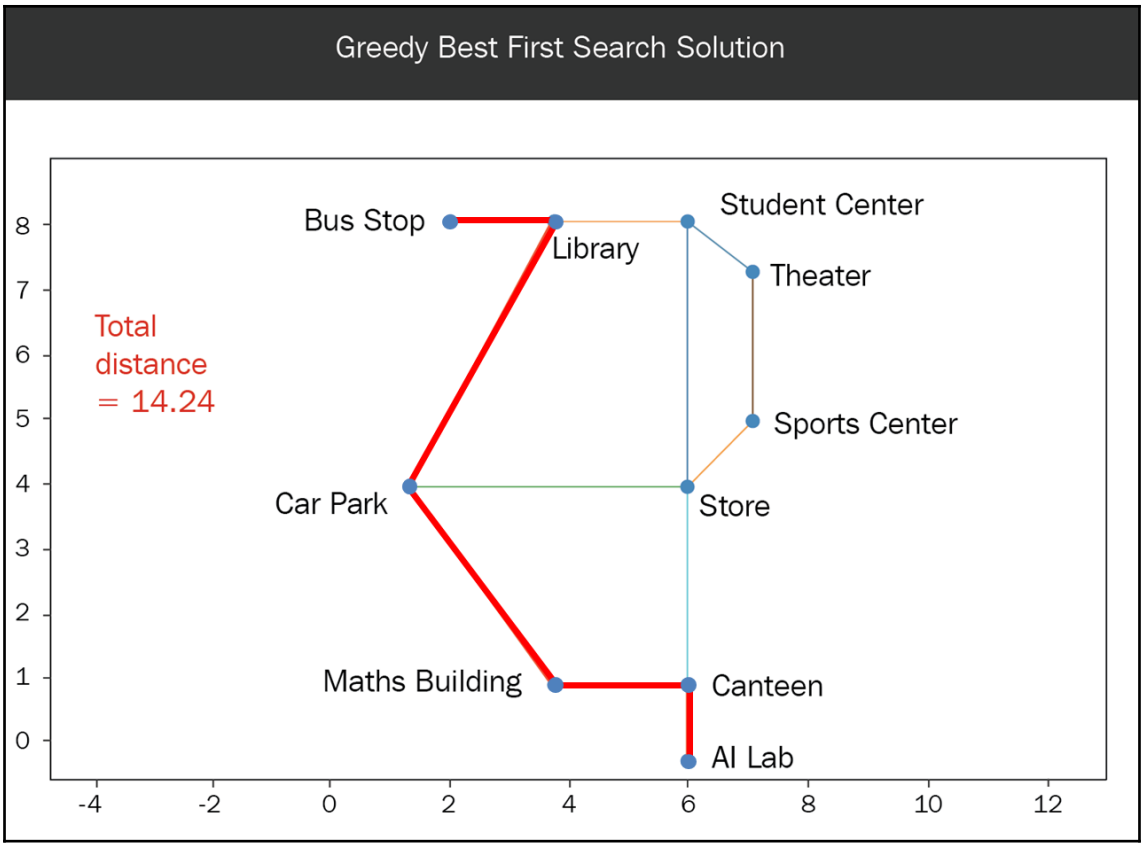

8.94427191
0 Bus Stop
8.24621125124
1 Library
8.0
6.40312423743
8.94427191
2 Student Center
3 Car Park
4 Bus Stop

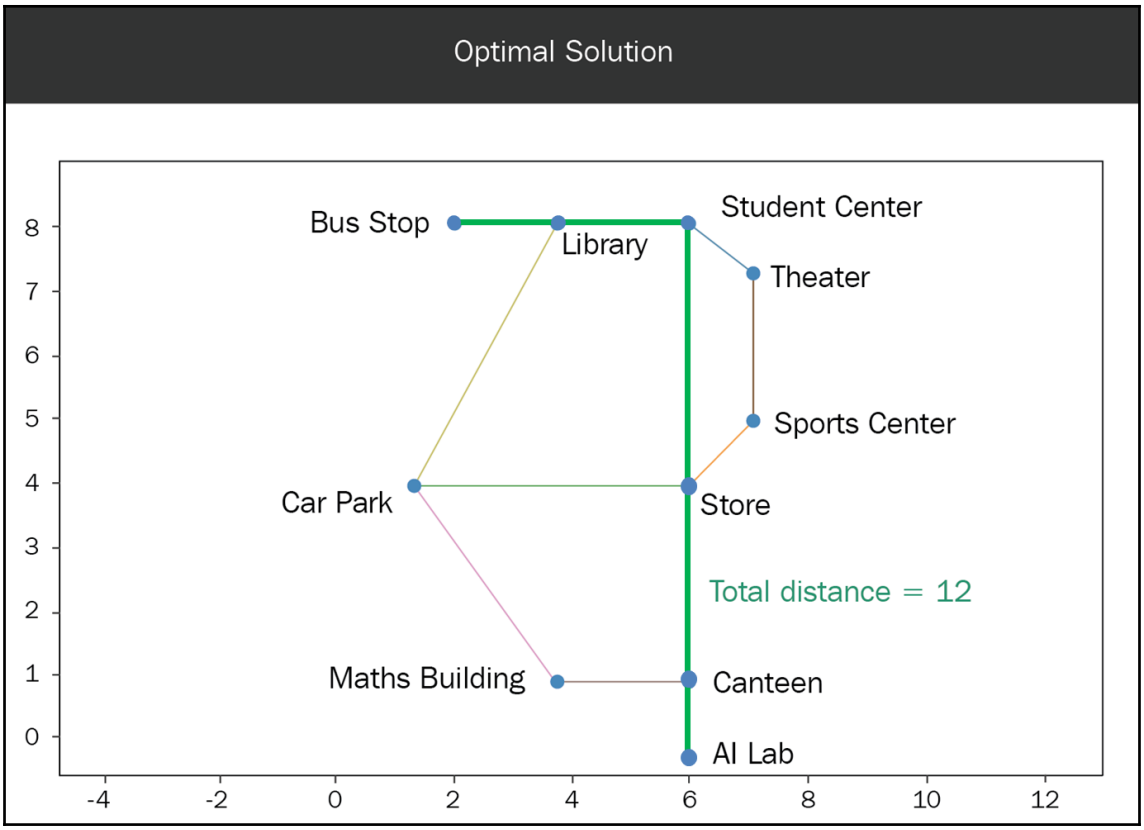

8.94427191
0 Bus Stop
8.24621125124
1 Library
8.0
6.40312423743
8.94427191
2 Student Center
3 Car Park
7 Bus Stop
2.2360679775
4.0
8.24621125124
4 Maths Building
5 Store
6 Library

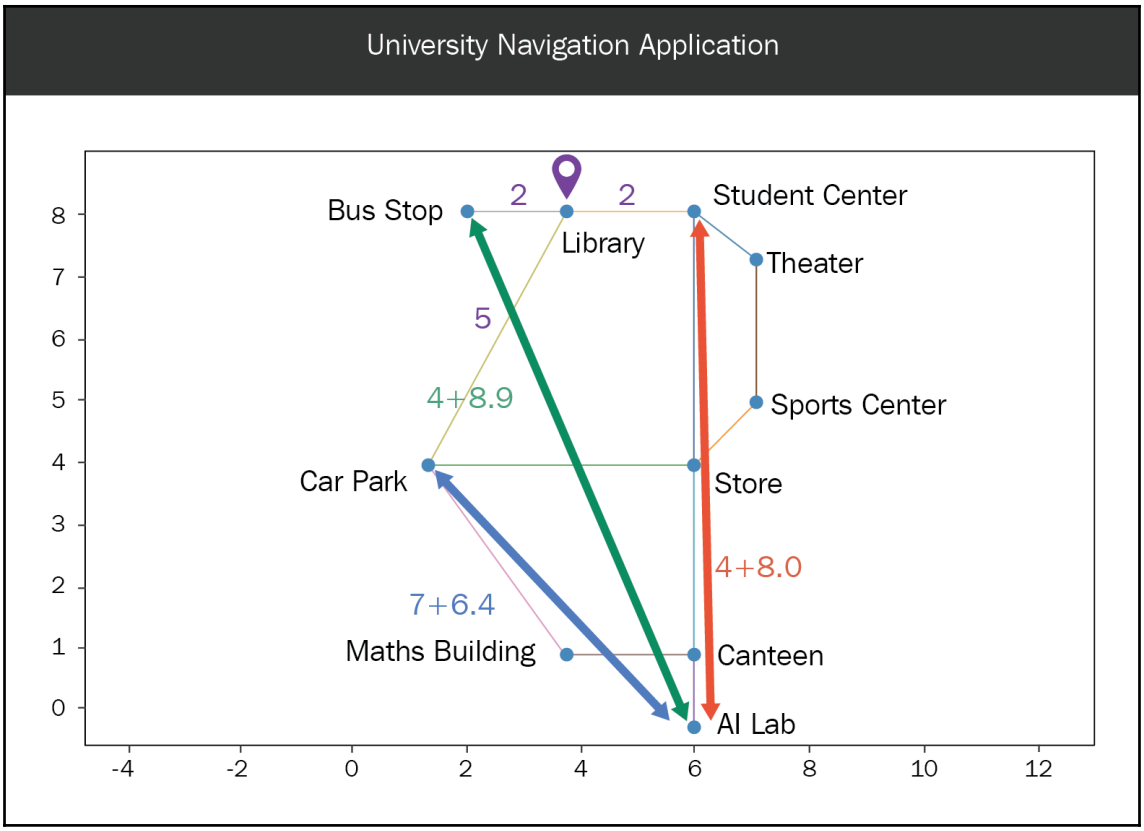

8.94427191
0 Bus Stop
8.24621125124
1 Library
8.0
6.40312423743
8.94427191
2 Student Center
3 Car Park
9 Bus Stop
2.2360679775
4.0
8.24621125124
4 Maths Building
7 Store
8 Library
1.0
6.40312423743
5 Canteen
6 Car Park

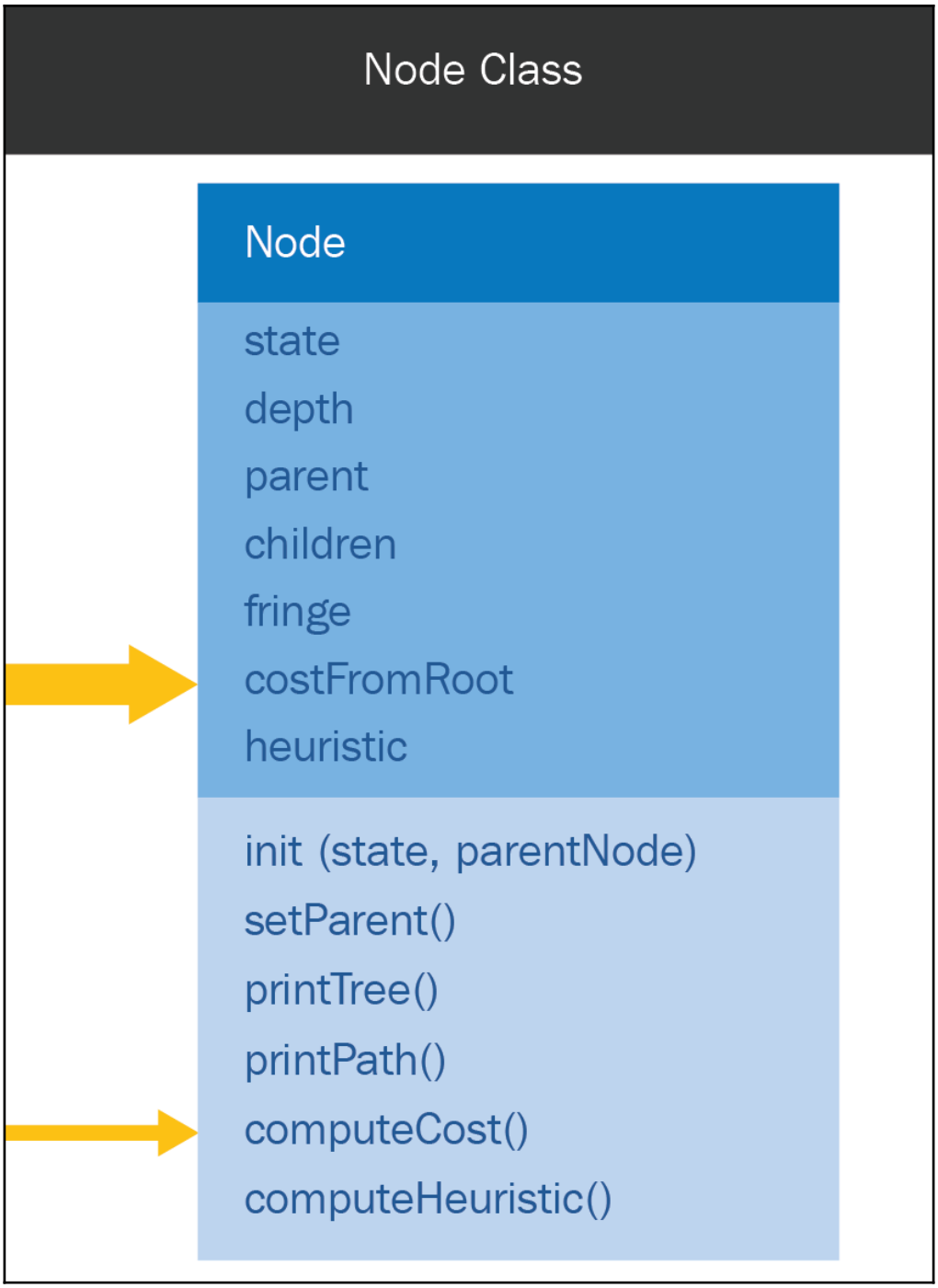

8.94427191
0 Bus Stop
8.24621125124
1 Library
8.0
6.40312423743
8.94427191
2 Student Center
3 Car Park
12 Bus Stop
2.2360679775
4.0
8.24621125124
4 Maths Building
10 Store
11 Library
1.0
6.40312423743
5 Canteen
6 Car Park
2.2360679775
0.0
4.0
6 Maths Building
7 AI Lab
8 Store


8.94427191
0 Bus Stop
8.24621125124
1 Library
8.0
6.40312423743
8.94427191
2 Student Center
3 Car Park
12 Bus Stop
2.2360679775
4.0
8.24621125124
4 Maths Building
10 Store
11 Library
1.0
6.40312423743
5 Canteen
9 Car Park
2.2360679775
0.0
4.0
6 Maths Building
7 AI Lab
8 Store

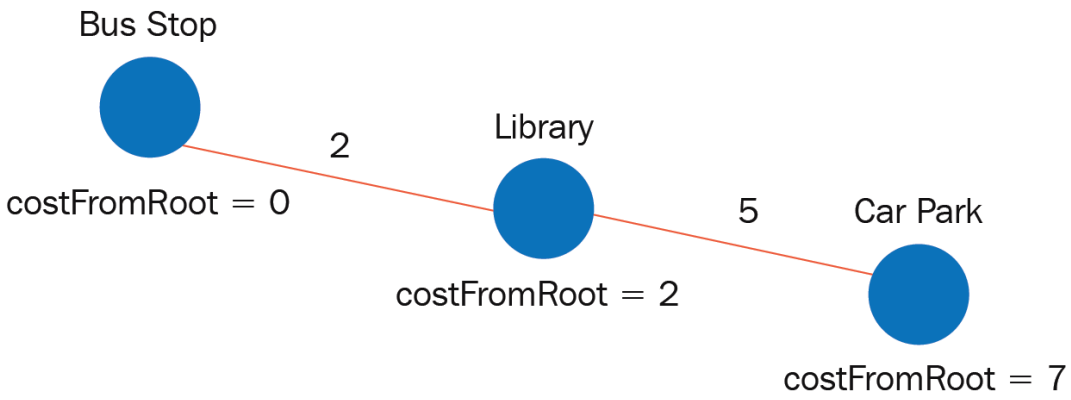

Greedy Best First Search Solution
Total
distance
= 14.24
Bus Stop
Library
Student Center
Theater
Sports Center
Car Park
Store
Maths Building
Canteen
AI Lab
8
7
6
5
4
3
2
1
0
-4
-2
0
2
4
6
8
10
12


Optimal Solution
Bus Stop
Library
Student Center
Theater
Sports Center
Car Park
Store
Total distance = 12
Maths Building
Canteen
AI Lab
8
7
6
5
4
3
2
1
0
-4
-2
0
2
4
6
8
10
12

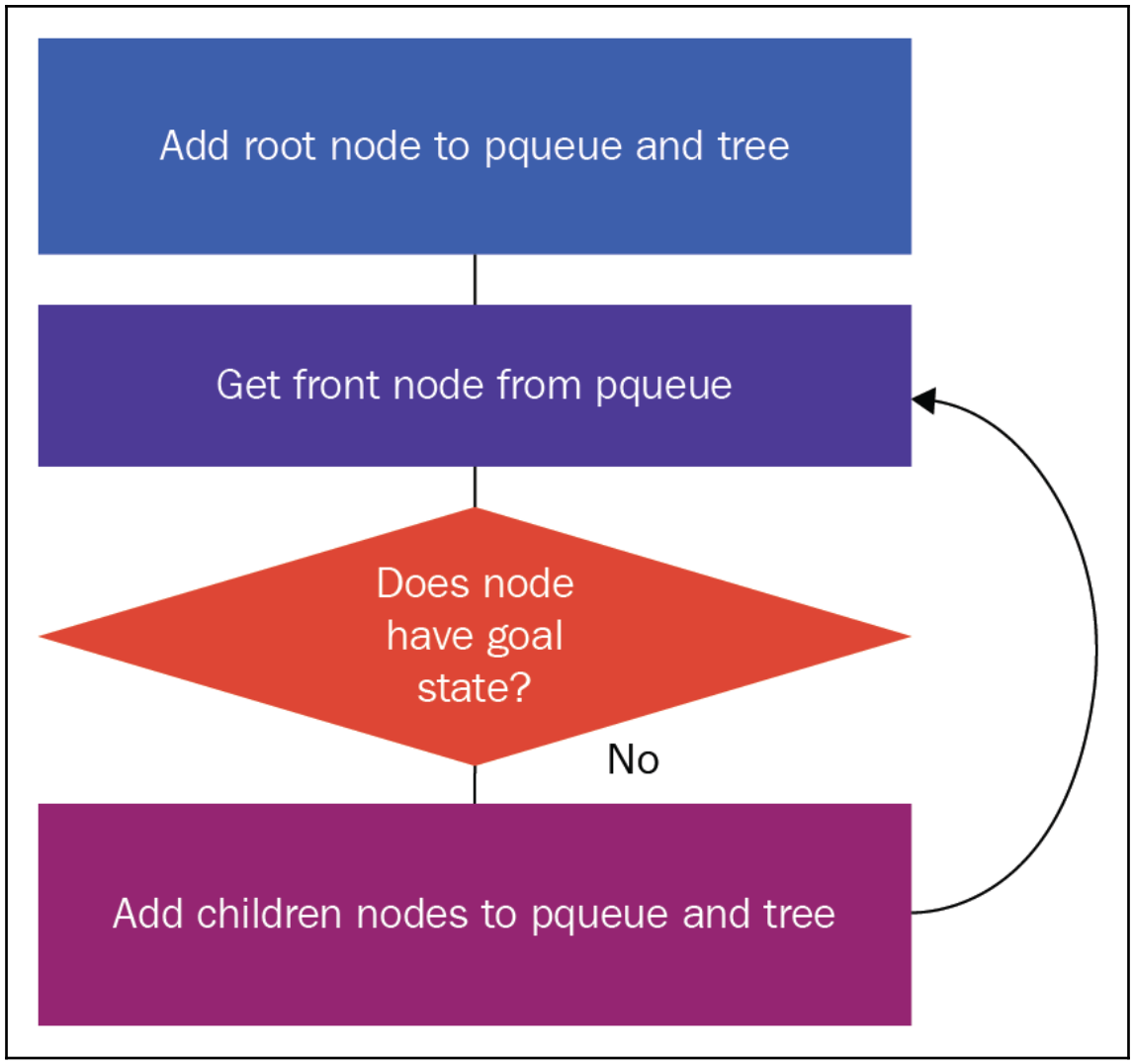

University Navigation Application
Bus Stop
Library
Student Center
Theater
Sports Center
Car Park
Store
Maths Building
Canteen
AI Lab
2
2
5
4+8.9
4+8.0
7+6.4
8
7
6
5
4
3
2
1
0
-4
-2
0
2
4
6
8
10
12

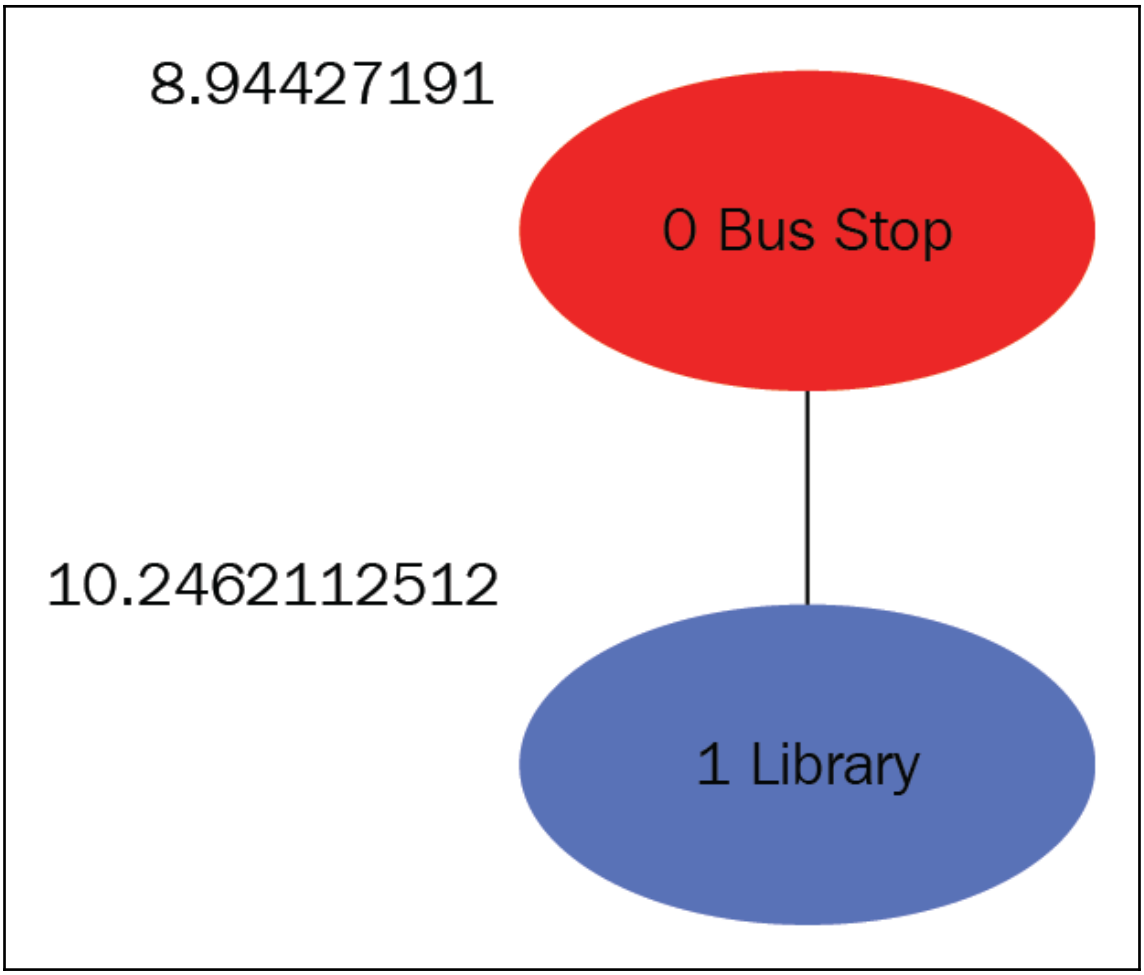

Node Class
Node
state
depth
parent
children
fringe
costFromRoot
heuristic
init (state, parentNode)
setParent()
printTree()
printPath()
computeCost()
computeHeuristic()

Node.computeCost()

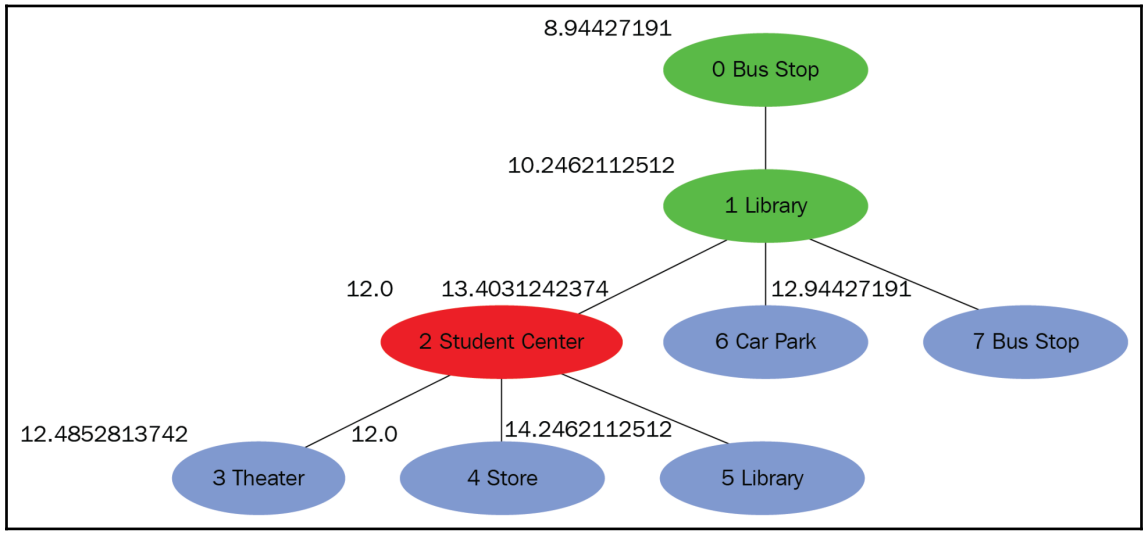


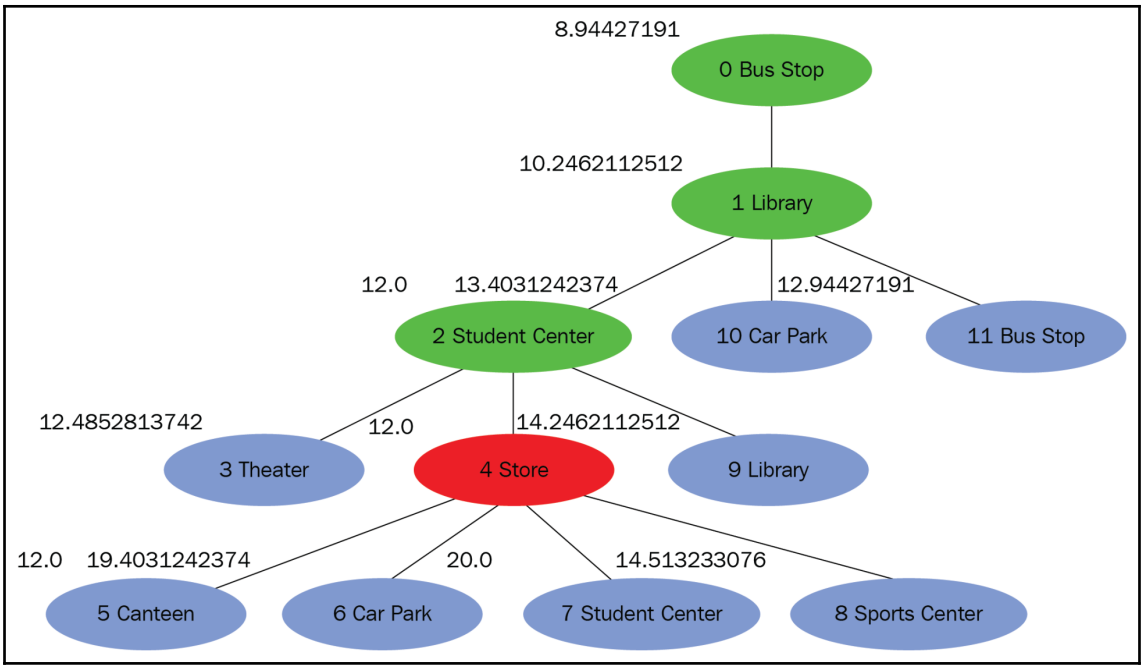
costFromRoot = parent's costFromRoot + distance of parent node to current node

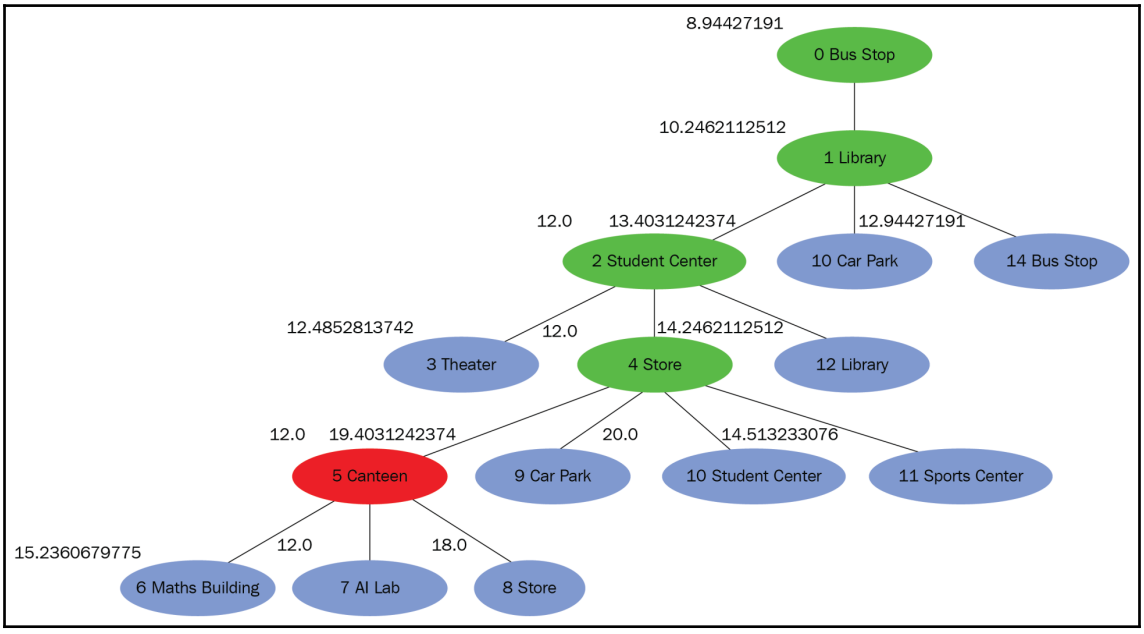

Add root node to pqueue and tree
Get front node from pqueue
Does node have goal state?
No
Add children nodes to pqueue and tree

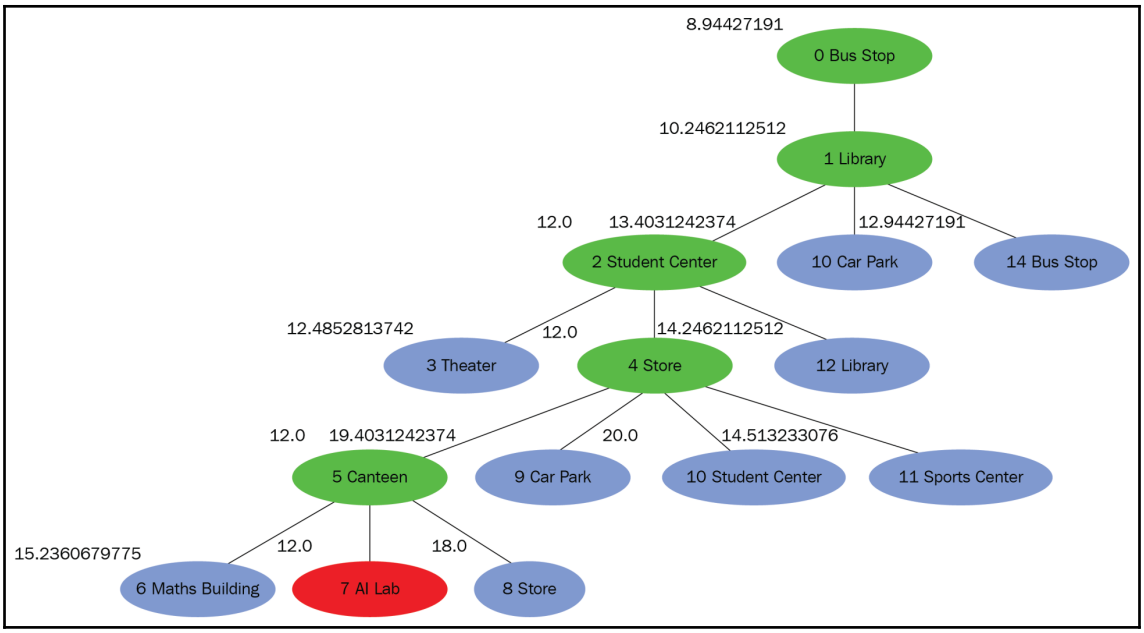

8.94427191
0 Bus Stop
10.2462112512
1 Library

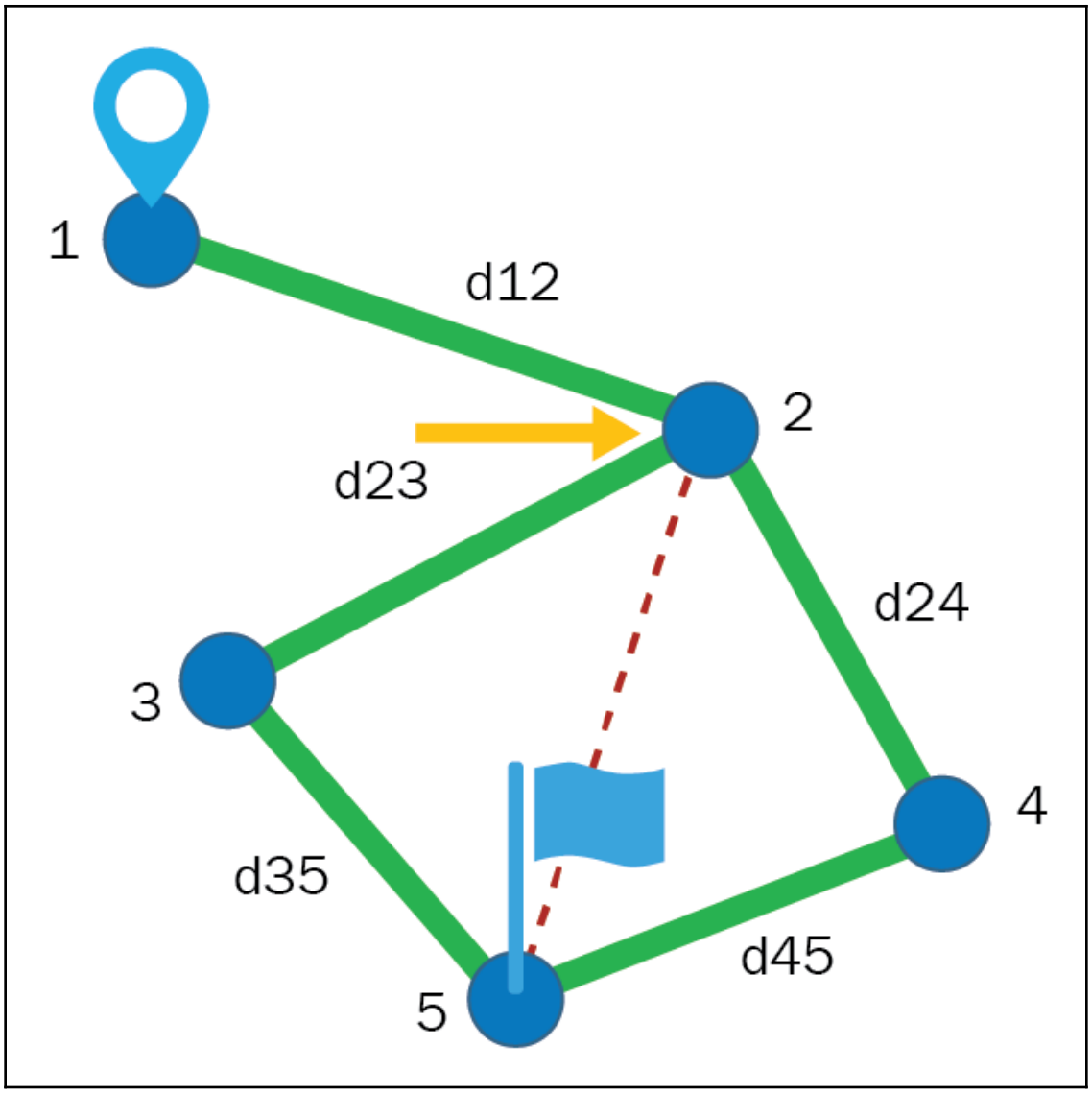

8.94427191
0 Bus Stop
10.2462112512
1 Library
12.94427191
13.4031242374
12.0
2 Bus Stop
3 Car Park
4 Student Center

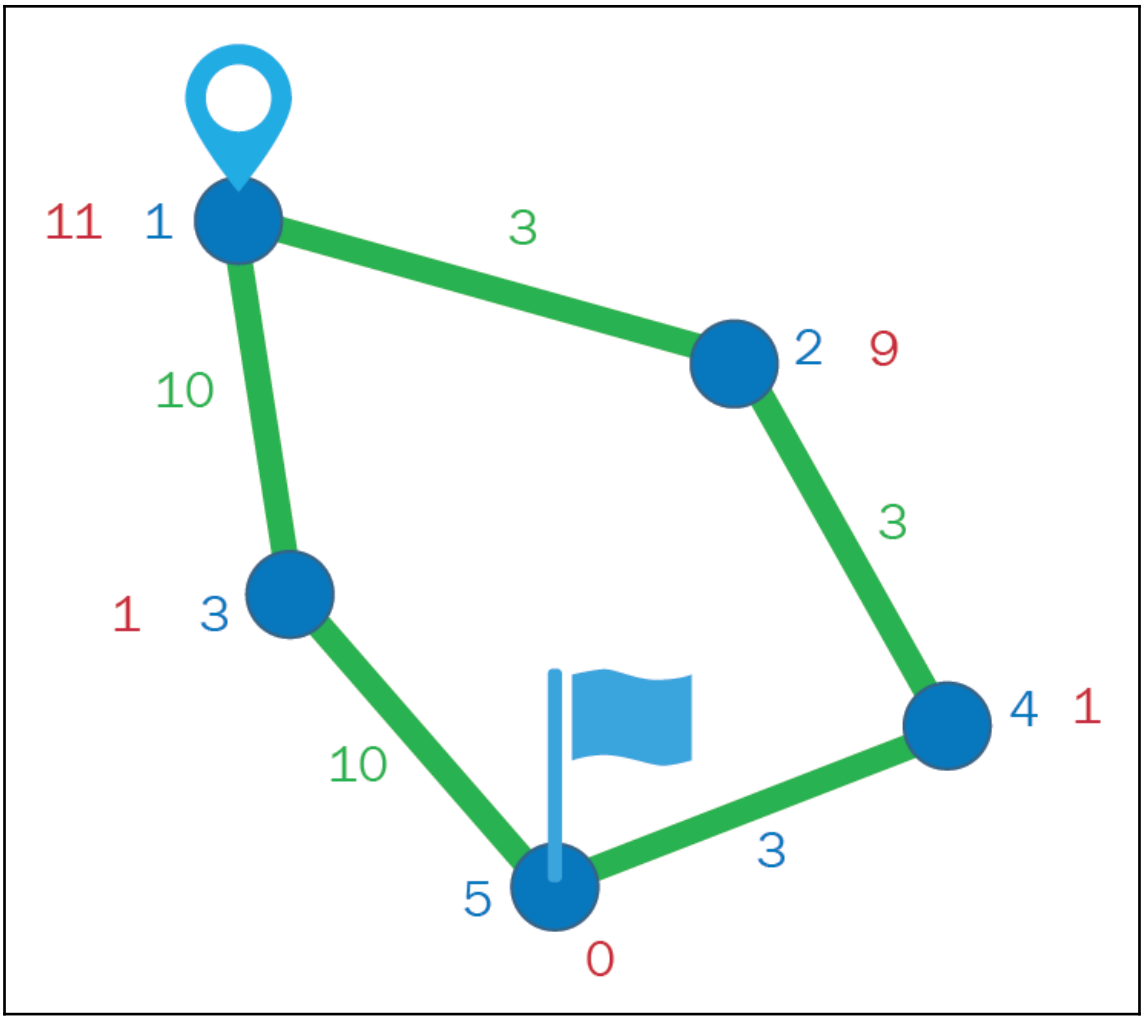

8.94427191
0 Bus Stop
10.2462112512
1 Library
12.0
13.4031242374
12.94427191
2 Student Center
6 Car Park
7 Bus Stop
12.4852813742
12.0
14.2462112512
3 Theater
4 Store
5 Library

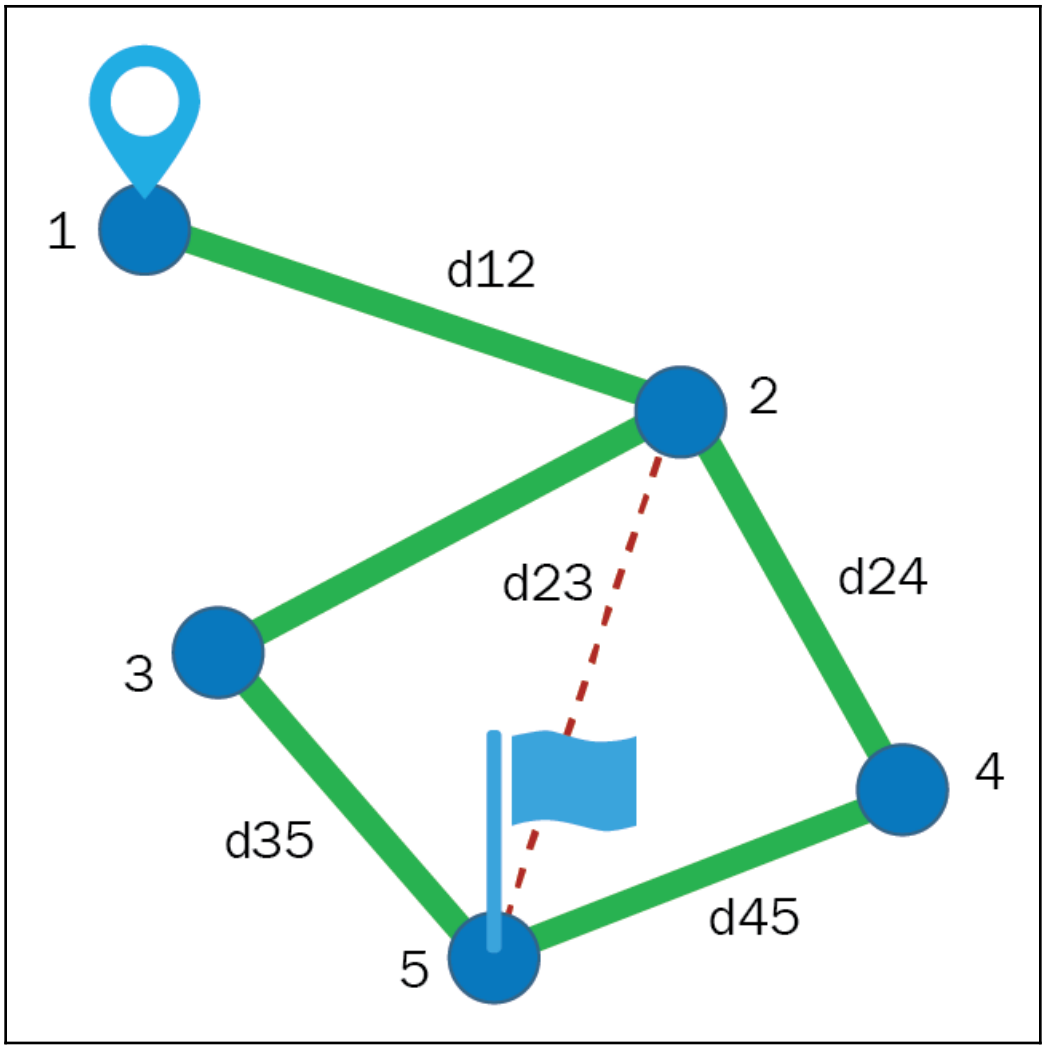

8.94427191
0 Bus Stop
10.2462112512
1 Library
12.0
13.4031242374
12.94427191
2 Student Center
10 Car Park
11 Bus Stop
12.4852813742
12.0
14.2462112512
3 Theater
4 Store
9 Library
12.0
19.4031242374
20.0
14.513233076
5 Canteen
6 Car Park
7 Student Center
8 Sports Center

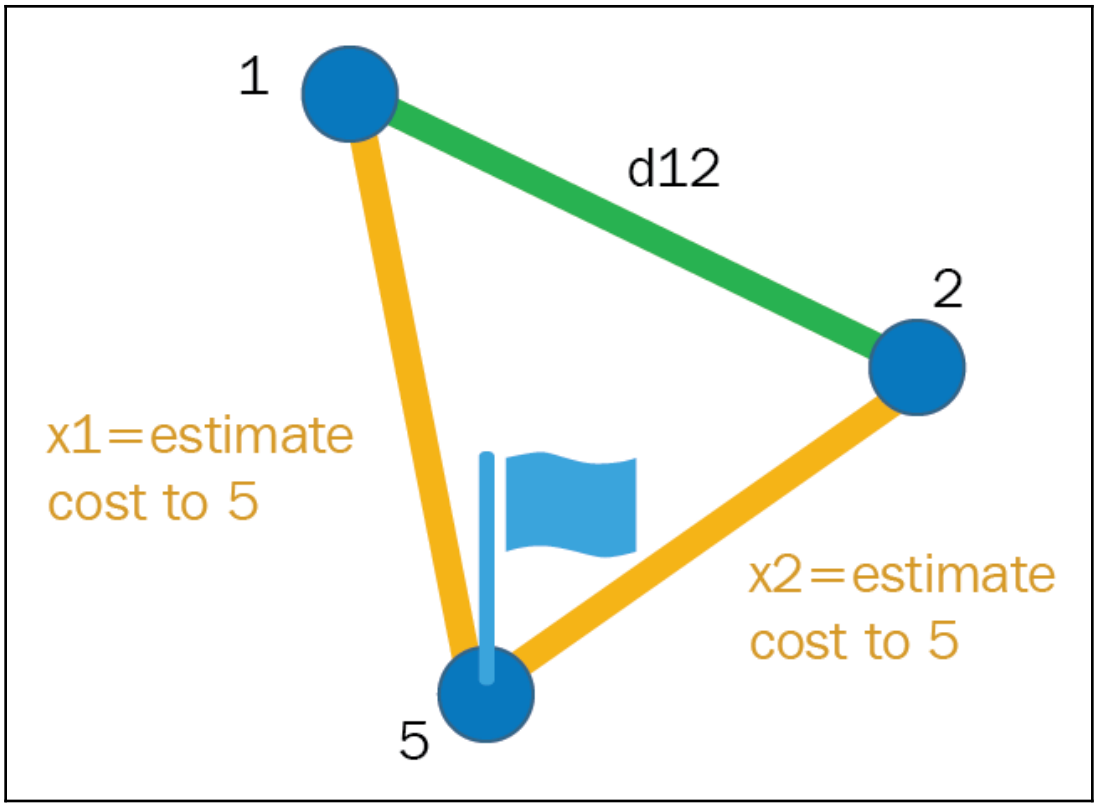

8.94427191
0 Bus Stop
10.2462112512
1 Library
12.0
13.4031242374
12.94427191
2 Student Center
10 Car Park
14 Bus Stop
12.4852813742
12.0
14.2462112512
3 Theater
4 Store
12 Library
12.0
19.4031242374
20.0
14.513233076
5 Canteen
9 Car Park
10 Student Center
11 Sports Center
15.2360679775
12.0
18.0
6 Maths Building
7 AI Lab
8 Store


8.94427191
0 Bus Stop
10.2462112512
1 Library
12.0
13.4031242374
12.94427191
2 Student Center
10 Car Park
14 Bus Stop
12.4852813742
12.0
14.2462112512
3 Theater
4 Store
12 Library
12.0
19.4031242374
20.0
14.513233076
5 Canteen
9 Car Park
10 Student Center
11 Sports Center
15.2360679775
12.0
18.0
6 Maths Building
7 AI Lab
8 Store


1
d12
2
d23
d24
3
4
d35
d45
5


11
1
3
2
9
10
3
1
3
4
1
10
3
5
0


1
d12
2
d23
d24
3
4
d35
d45
5


1
d12
2
x1=estimate
cost to 5
x2=estimate
cost to 5
5