SCC461 – Programming for Data Scientists

Leandro Marcolino

Lancaster University

Week 10

Outline

1. Private Members
2. Peer Feedback
3. DS Libraries
4. Problem Solving
5. Assignment
6. Conclusion

Stack again...

```
class Stack:
     def __init__(self):
         self.items = 10*[0];
         self.position = 0;

     def push(self, item):
          if (self.position < 10):
              self.items[self.position] = item;
              self.position = self.position + 1;
              return True;
          else:
              return False;
```

...

Stack again...

```
    def pop(self):
        if (self.position <= 0):
            return False;
        else:
            self.position = self.position - 1;
            return self.items[self.position];

stack = Stack();

stack.push(5);
stack.push(10);
stack.pop();
stack.pop();
print(stack.items[1]);
```

Stack again...

```
    def pop(self):
        if (self.position <= 0):
            return False;
        else:
            self.position = self.position - 1;
            return self.items[self.position];

stack = Stack();

stack.push(5);
stack.push(10);
stack.pop();
stack.pop();
print(stack.items[1]);
```

Why this code is wrong?

Private Members

```
class Stack:
     def __init__(self):
         self._items = 10*[0];
         self._position = 0;

     def push(self, item):
          if (self._position < 10):
              self._items[self._position] = item;
              self._position = self._position +
                 1;
              return True;
          else:
              return False;
```

...

Stack again...

```
def pop(self):
    if (self._position <= 0):
        return False;
    else:
        self._position = self._position - 1;
        return self._items[self.position];

def _checkStack(self):
    ....
```

Outline

1. Private Members
2. Peer Feedback
3. DS Libraries
4. Problem Solving
5. Assignment
6. Conclusion

Peer Feedback Exercise

- WAIT FOR ALL INSTRUCTIONS BEFORE YOU START MOVING!
- You will work in pairs
- Discuss your CW 8 and 9 with your partner
- Ask your partner what he/she is struggling with, and teach him/her
- Similarly, tell your partner what you are struggling with, and he/she will teach you
- If you are new to programming, find an experienced programmer as a pair
- If you are an experienced programmer, find someone that is new to programming as a pair

Outline

1. Private Members
2. Peer Feedback
3. DS Libraries
4. Problem Solving
5. Assignment
6. Conclusion

SciPy

NumPy

Base N-dimensional array package

SciPy library

Fundamental library for scientific computing

Matplotlib

Comprehensive 2D Plotting

IPython

Enhanced Interactive Console

Sympy

Symbolic mathematics

pandas

Data structures & analysis

NumPy

Basic Element: Matrix (homogeneous multidimensional array)

$$\begin{bmatrix} 4 & 5 & 8 & 3 \\ 10 & 3 & -1 & 0 \\ 5 & 7 & 4 & 5 \end{bmatrix}$$

- Number of dimensions: 2
- Shape: (3, 4). 3 Rows, 4 Columns.

NumPy

Array Creation

```
import numpy as np

a = np.array([2,3,4])
```

$$a = \begin{bmatrix} 2 & 3 & 4 \end{bmatrix}$$

NumPy

Array Creation

```
b = np.array([(1.5,2,3), (4,5,6)])
```

$$b = \begin{bmatrix} 1.5 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}$$

NumPy

Array Creation

```
a = np.zeros((3,4))
```

$$a = \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

```
a = np.ones((3,4))
```

$$a = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

NumPy

Array Creation

```
a = np.ones((2,3,4))
```

$$a = \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \\ 1 & 1 & 1 & 1 \end{bmatrix}$$

NumPy

Array Creation

```
a = np.arrange(6)
```

$$a = \begin{bmatrix} 0 & 1 & 2 & 3 & 4 & 5 \end{bmatrix}$$

```
b = np.arange(12).reshape(4,3)
```

$$a = \begin{bmatrix} 0 & 1 & 2 \\ 3 & 4 & 5 \\ 6 & 7 & 8 \\ 9 & 10 & 11 \end{bmatrix}$$

Basic Operations

```
a = np.array( [20,30,40,50] )
b = np.arange( 4 )

c = a - b
```

$$c = \begin{bmatrix} 20 & 29 & 38 & 47 \end{bmatrix}$$

Basic Operations

```
b = np.arange( 4 )

a = b**2
```

$$a = \begin{bmatrix} 0 & 1 & 4 & 9 \end{bmatrix}$$

Basic Operations

```
A = np.array([[1,1],
              [0,1]])
B = np.array([[2,0],
               [3,4]])
a = A*B
```

$$a = \begin{bmatrix} 2 & 0 \\ 0 & 4 \end{bmatrix}$$

```
a = A.dot(B)
```

$$a = \begin{bmatrix} 5 & 4 \\ 3 & 4 \end{bmatrix}$$

Basic Operations

```
a = np.random.random((2,3))
```

$$a = \begin{bmatrix} 0.18626021 & 0.34556073 & 0.39676747 \\ 0.53881673 & 0.41919451 & 0.6852195 \end{bmatrix}$$

```
a.sum()
```

2.5718191614547998

```
a.min()
```

0.1862602113776709

```
a.max()
```

0.6852195003967595

Indexing

```
def f(x,y):
    return 10*x+y
```

```
b = np.fromfunction(f,(5,4),dtype=int)
```

$$b = \begin{bmatrix} 0 & 1 & 2 & 3 \\ 10 & 11 & 12 & 13 \\ 20 & 21 & 22 & 23 \\ 30 & 31 & 32 & 33 \\ 40 & 41 & 42 & 43 \end{bmatrix}$$

```
b[2,3]
```

23

```
b[0:5, 1]
```

$$\begin{bmatrix} 1 & 11 & 21 & 31 & 41 \end{bmatrix}$$

SciPy

Subpackage	Description
cluster	Clustering algorithms
constants	Physical and mathematical constants
fftpack	Fast Fourier Transform routines
integrate	Integration and ordinary differential equation solvers
interpolate	Interpolation and smoothing splines
io	Input and Output
linalg	Linear algebra
ndimage	N-dimensional image processing
odr	Orthogonal distance regression
optimize	Optimization and root-finding routines
signal	Signal processing
sparse	Sparse matrices and associated routines
spatial	Spatial data structures and algorithms
special	Special functions
stats	Statistical distributions and functions

SciPy

Optimization

Rosenbrock Function:

$$f(\mathbf{x}) = \sum_{i=1}^{N-1} 100(x_i - x_{i-1}^2)^2 + (1 - x_{i-1})^2$$

SciPy

Optimization

```
import numpy as np
from scipy.optimize import minimize

def rosen(x):
    return sum(100.0*(x[1:]-x[:-1]**2.0)**2.0 +
        (1-x[:-1])**2.0)

x0 = np.array([1.3, 0.7, 0.8, 1.9, 1.2])
res = minimize(rosen, x0, method='nelder-mead',
               options={'xtol': 1e-8, 'disp':
                  True})
```

```
Optimization terminated successfully.
Current function value: 0.000000
Iterations: 339
```

SciPy

Interpolation

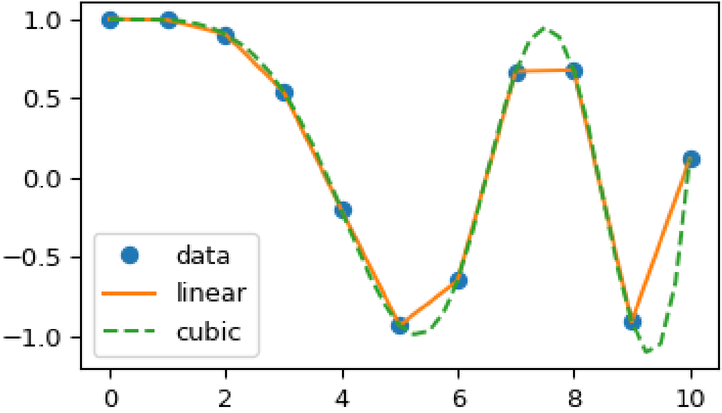
```
from scipy.interpolate import interp1d
import matplotlib.pyplot as plt

x = np.linspace(0, 10, num=11, endpoint=True)
y = np.cos(-x**2/9.0)
f = interp1d(x, y)
f2 = interp1d(x, y, kind='cubic')

xnew = np.linspace(0, 10, num=41, endpoint=True)
plt.plot(x, y, 'o', xnew, f(xnew), '-', xnew,
   f2(xnew), '--')
plt.legend(['data', 'linear', 'cubic'],
   loc='best')
plt.show()
```

SciPy

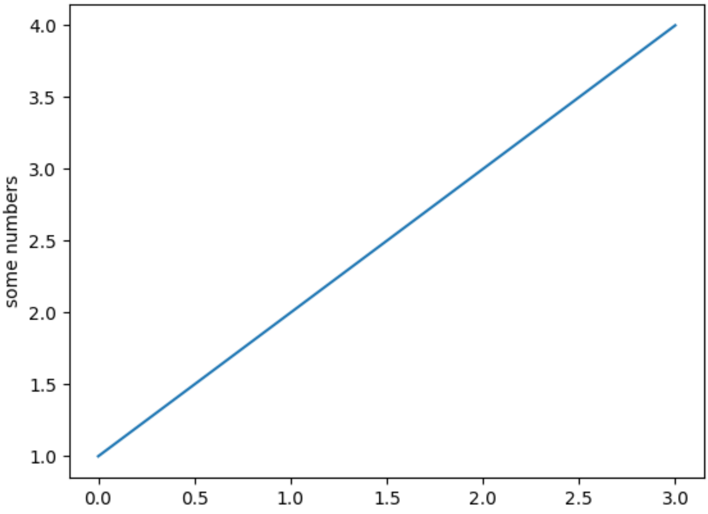
Interpolation



PyPlot

```
import matplotlib.pyplot as plt
plt.plot([1, 2, 3, 4])
plt.ylabel('some numbers')
plt.show()
```

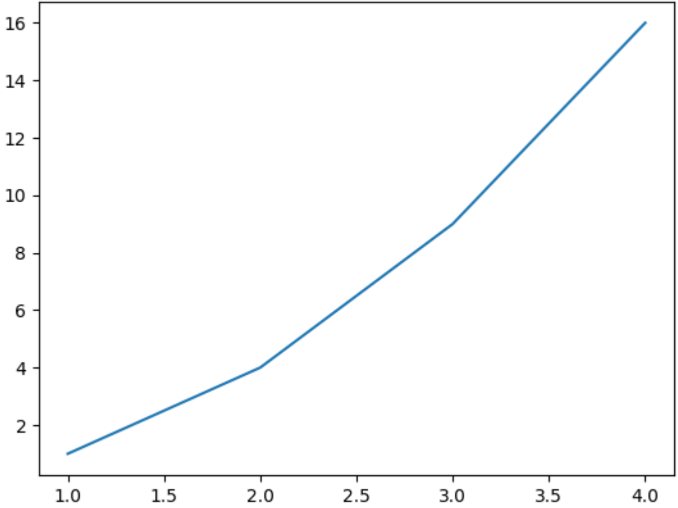
PyPlot



PyPlot

```
plt.plot([1, 2, 3, 4], [1, 4, 9, 16])
plt.show()
```

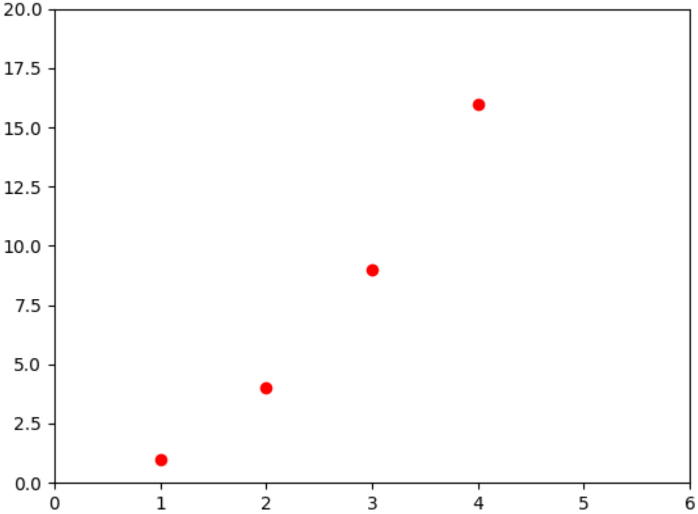
PyPlot



PyPlot

```
plt.plot([1, 2, 3, 4], [1, 4, 9, 16], 'ro')
plt.axis([0, 6, 0, 20])
plt.show()
```

PyPlot



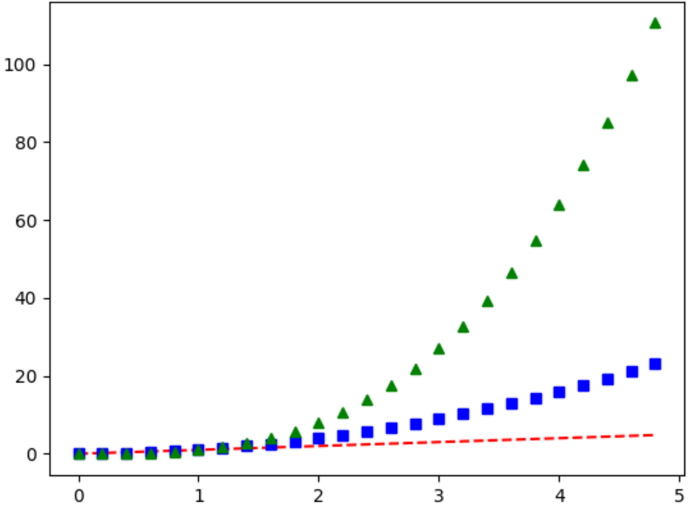
PyPlot

```
import numpy as np

# evenly sampled time at 200ms intervals
t = np.arange(0., 5., 0.2)

# red dashes, blue squares and green triangles
plt.plot(t, t, 'r--', t, t**2, 'bs', t, t**3,
    'g^')
plt.show()
```

PyPlot



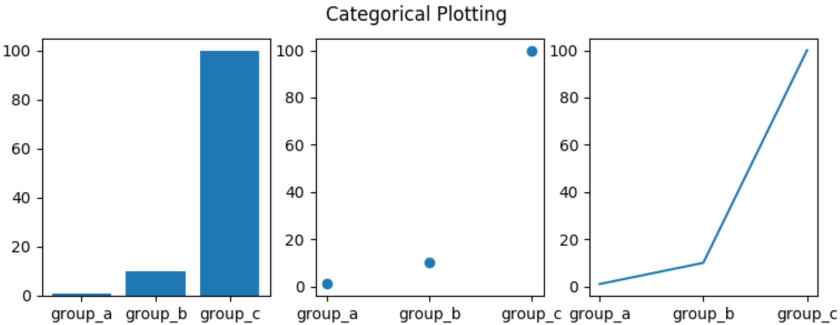
PyPlot

```
names = [’group_a’, ’group_b’, ’group_c’]
values = [1, 10, 100]

plt.figure(1, figsize=(9, 3))

plt.subplot(131)
plt.bar(names, values)
plt.subplot(132)
plt.scatter(names, values)
plt.subplot(133)
plt.plot(names, values)
plt.suptitle(’Categorical Plotting’)
plt.show()
```

PyPlot


Categorical Plotting
100
80
60
40
20
0
group_a
group_b
group_c
100
80
60
40
20
0
group_a
group_b
group_c
100
80
60
40
20
0
group_a
group_b
group_c

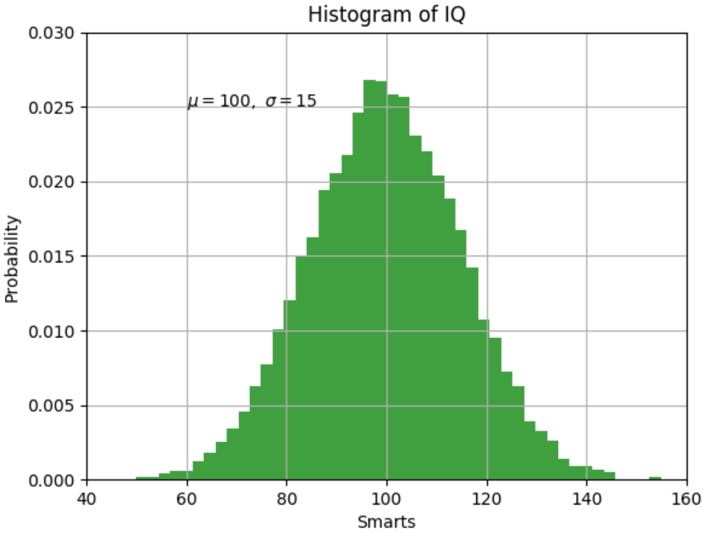
PyPlot

```
mu, sigma = 100, 15
x = mu + sigma * np.random.randn(10000)

# the histogram of the data
n, bins, patches = plt.hist(x, 50, normed=1,
   facecolor='g', alpha=0.75)

plt.xlabel('Smarts')
plt.ylabel('Probability')
plt.title('Histogram of IQ')
plt.text(60, .025, r'$\mu=100,\ \sigma=15$')
plt.axis([40, 160, 0, 0.03])
plt.grid(True)
plt.show()
```

PyPlot


Histogram of IQ
$\mu = 100,\ \sigma = 15$
Probability
0.030
0.025
0.020
0.015
0.010
0.005
0.000
40
60
80
100
120
140
160
Smarts

More Information

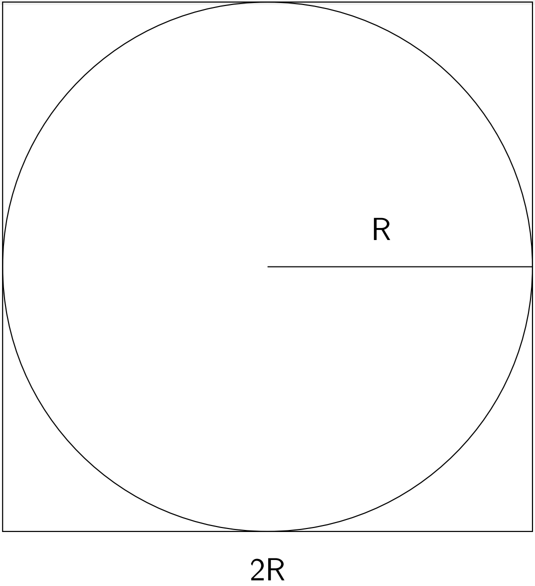
- NumPy: http://www.numpy.org/
- SciPy: https://www.scipy.org/
- Matplotlib: http://matplotlib.org/

Must see! :D

Outline

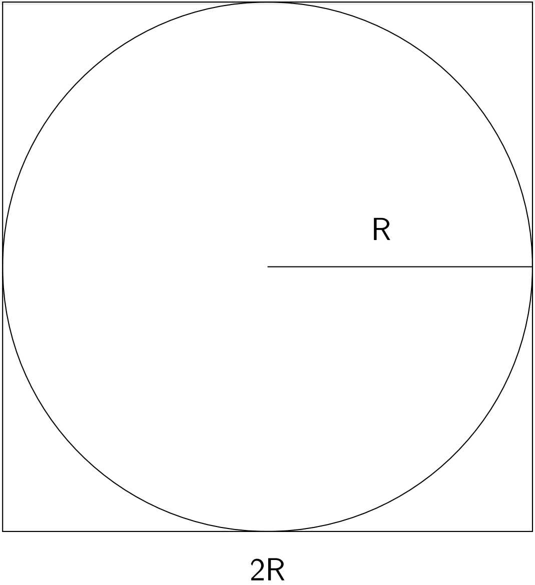
1. Private Members
2. Peer Feedback
3. DS Libraries
4. Problem Solving
5. Assignment
6. Conclusion

Monte Carlo Method



Area of circle:
$\pi \times R^2$

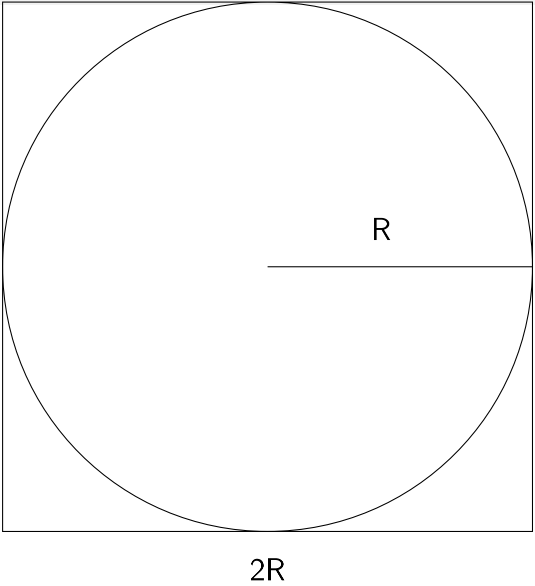
Monte Carlo Method



Area of circle:
$\pi \times R^2$

Area of square:
$(2R)^2$

Monte Carlo Method

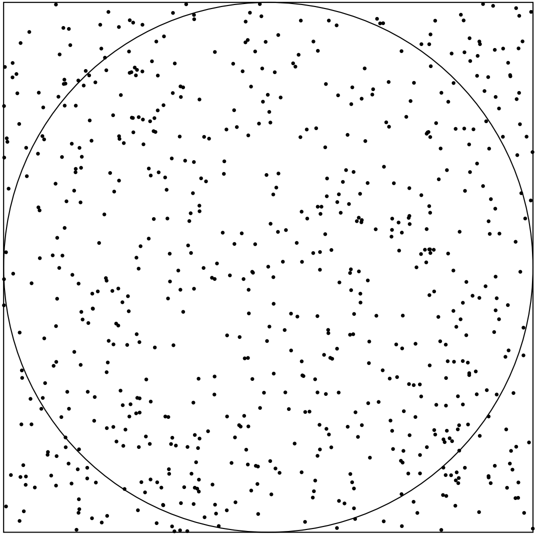


Area of circle:
$\pi \times R^2$

Area of square:
$(2R)^2$

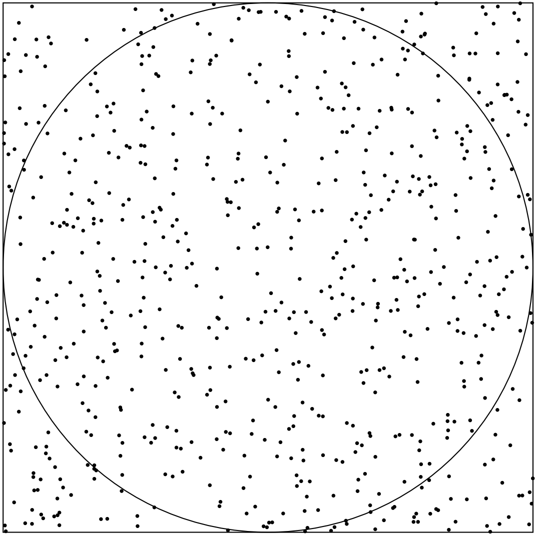
Ratio:
$\frac{\pi}{4}$

Monte Carlo Method



N random points inside the square

Monte Carlo Method

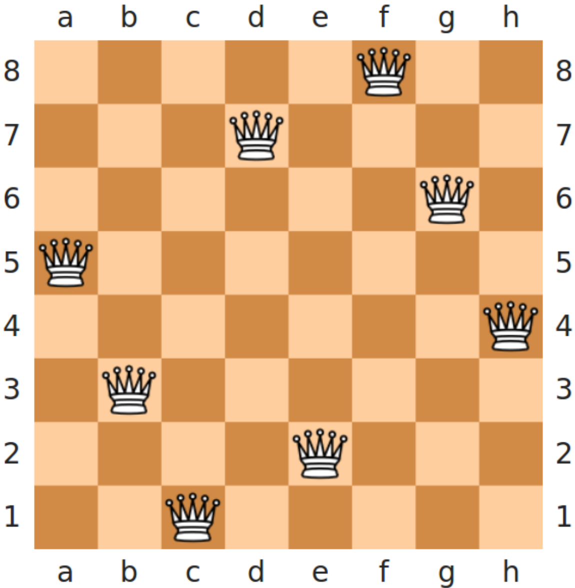


N random points inside the square

Inside circle:

$$\frac{N*\pi}{4}$$

8 Queens Problem



Backtracking

- Place the 1st queen on the board.
- Place the 2nd queen on the board. If not safe, change position of 2nd queen.
- ...
- Place queen n on the board. If not safe, change position of queen n.
- If we ran out of positions for queen n, find a new position for queen $n - 1$.
- If we ran out of positions for queen $n - 1$, find a new position for queen $n - 2$...

Backtracking

- Place the 1st queen on the board.
- Place the 2nd queen on the board. If not safe, change position of 2nd queen.
- ...
- Place queen n on the board. If not safe, change position of queen n.
- If we ran out of positions for queen n, find a new position for queen $n - 1$.
- If we ran out of positions for queen $n - 1$, find a new position for queen $n - 2$...

Can be implemented with:
a stack! :)

Backtracking

- Place the 1st queen on the board.
- Place the 2nd queen on the board. If not safe, change position of 2nd queen.
- ...
- Place queen n on the board. If not safe, change position of queen n.
- If we ran out of positions for queen n, find a new position for queen $n - 1$.
- If we ran out of positions for queen $n - 1$, find a new position for queen $n - 2$...

Can be implemented with: a stack! :)

Can be implemented with: recursive functions!

Backtracking

Animation at: https://www.youtube.com/watch?v=ckC2hFdLff0

Many resources at:

https://www.youtube.com/results?search_query=8+queen+puzzle+backtracking

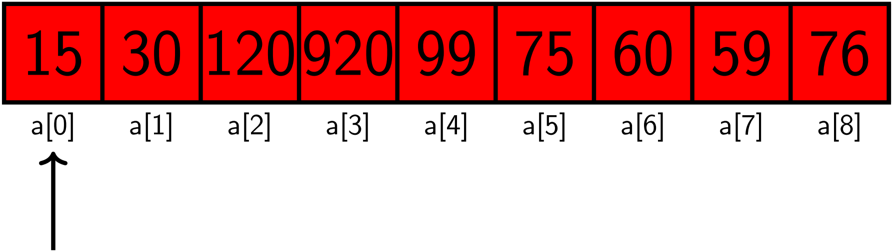
Linear Search

Is 99 on the list?

15	30	120	920	99	75	60	59	76
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]

Linear Search

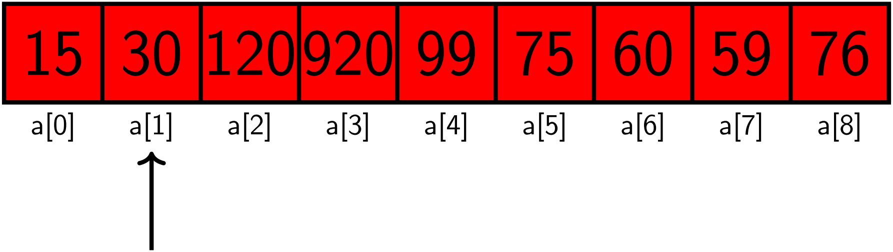
Is 99 on the list?



15	30	120	920	99	75	60	59	76
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]

Linear Search

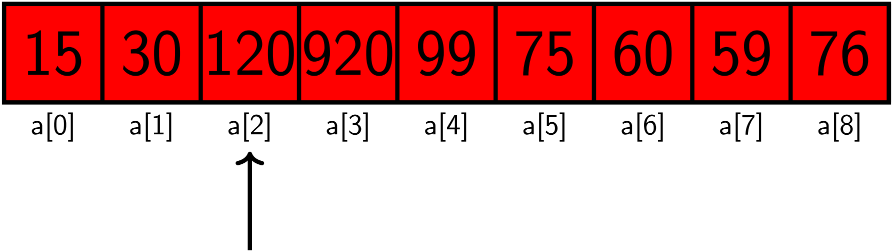
Is 99 on the list?



15	30	120	920	99	75	60	59	76
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]

Linear Search

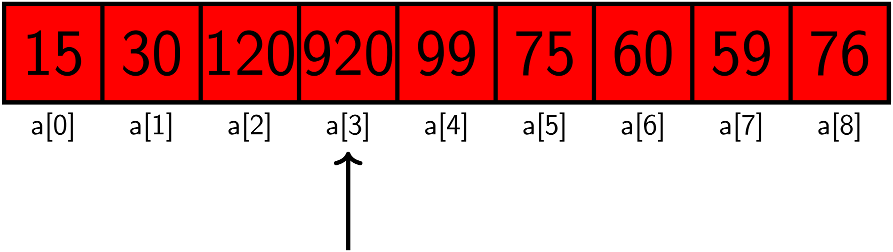
Is 99 on the list?



15	30	120	920	99	75	60	59	76
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]

Linear Search

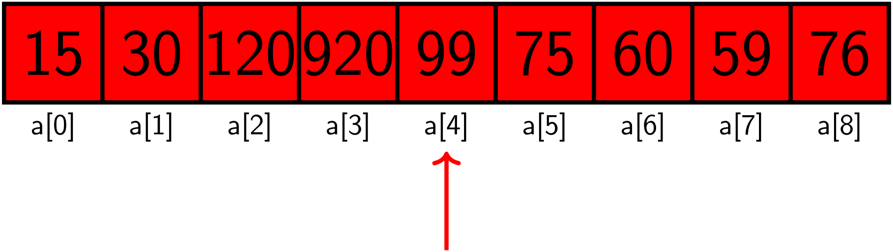
Is 99 on the list?



15	30	120	920	99	75	60	59	76
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]

Linear Search

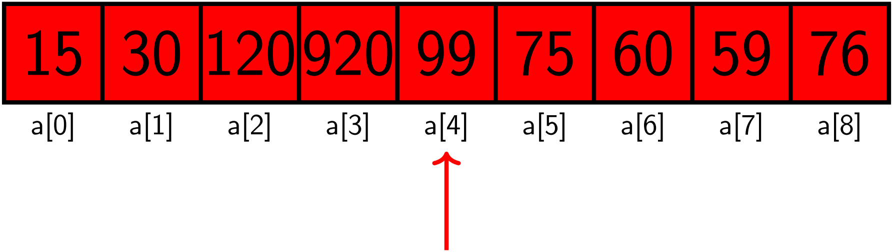
Is 99 on the list?



15	30	120	920	99	75	60	59	76
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]

Linear Search

Is 99 on the list?

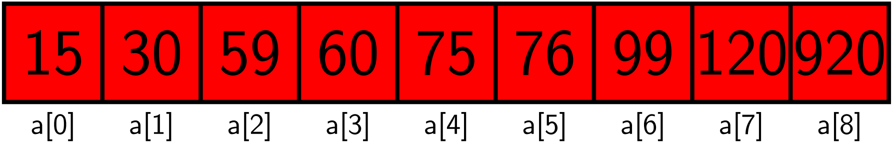


a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]
15	30	120	920	99	75	60	59	76

Can we do better?

Binary Search

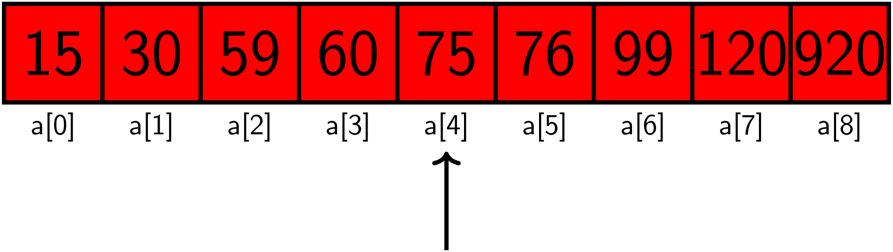
Is 99 on the list?



a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]
15	30	59	60	75	76	99	120	920

Binary Search

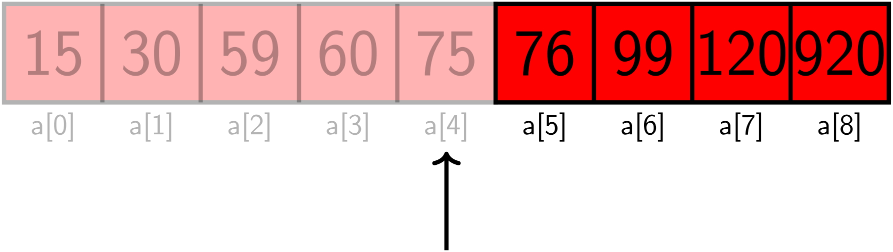
Is 99 on the list?



15	30	59	60	75	76	99	120	920
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]

Binary Search

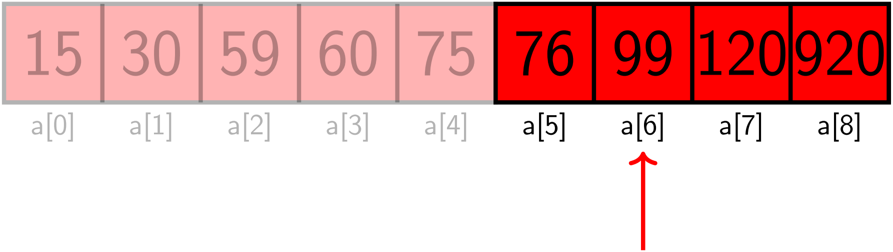
Is 99 on the list?



15	30	59	60	75	76	99	120	920
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]

Binary Search

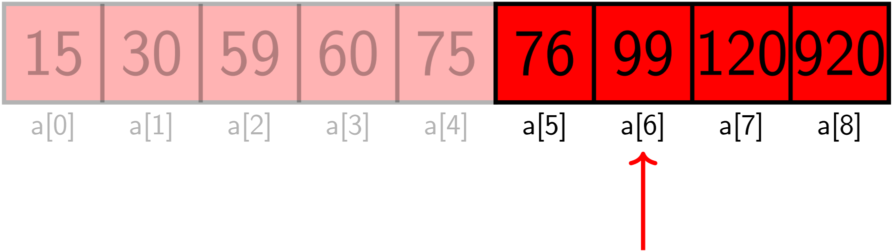
Is 99 on the list?



15	30	59	60	75	76	99	120	920
a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]

Binary Search

Is 99 on the list?



a[0]	a[1]	a[2]	a[3]	a[4]	a[5]	a[6]	a[7]	a[8]
15	30	59	60	75	76	99	120	920

Divide and Conquer!

Binary Search

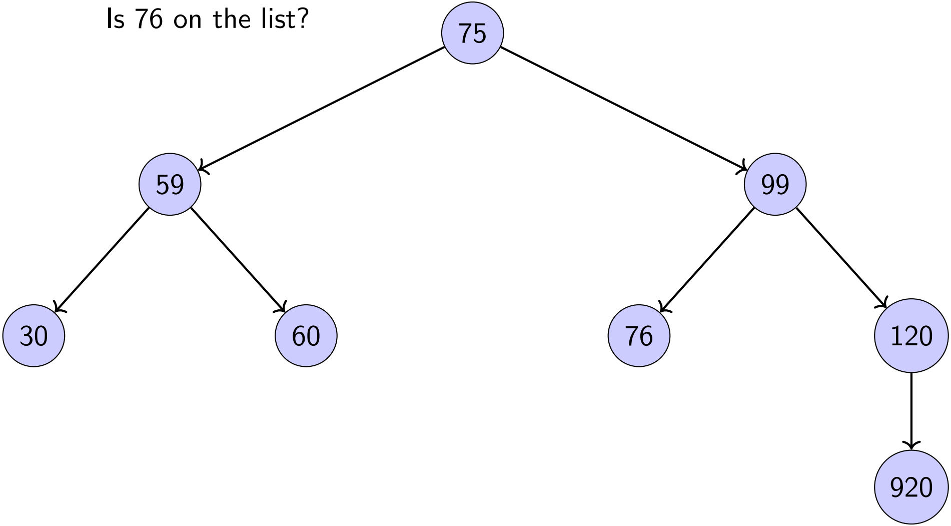
```
def binary_search(myList, item, low, high):
    if (high < low):
        return -1;

    midPoint = (low + high) // 2

    if (item == myList[midPoint]):
        return midPoint
    elif (item < myList[midPoint]):
        return binary_search(myList, item, low,
           mid - 1)
    else:
        return binary_search(myList, item, mid +
           1, high)
```

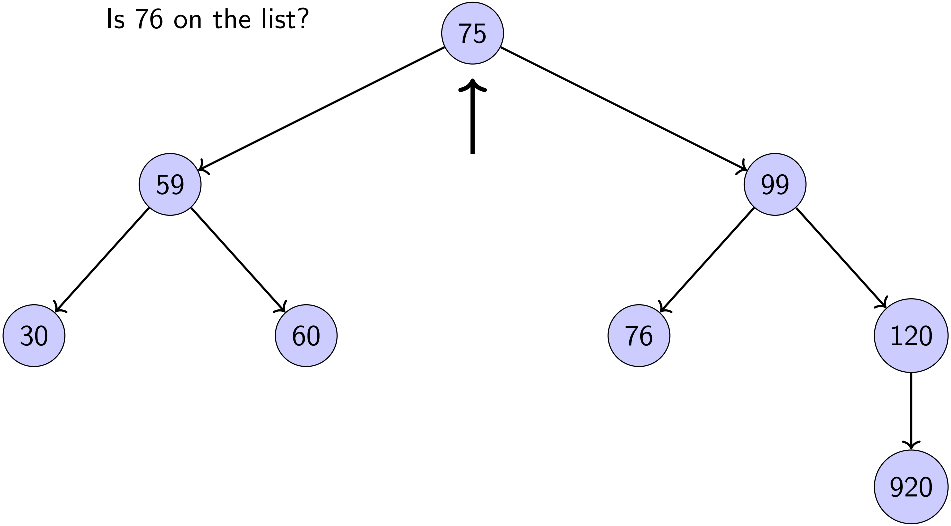
Search Tree

Is 76 on the list?



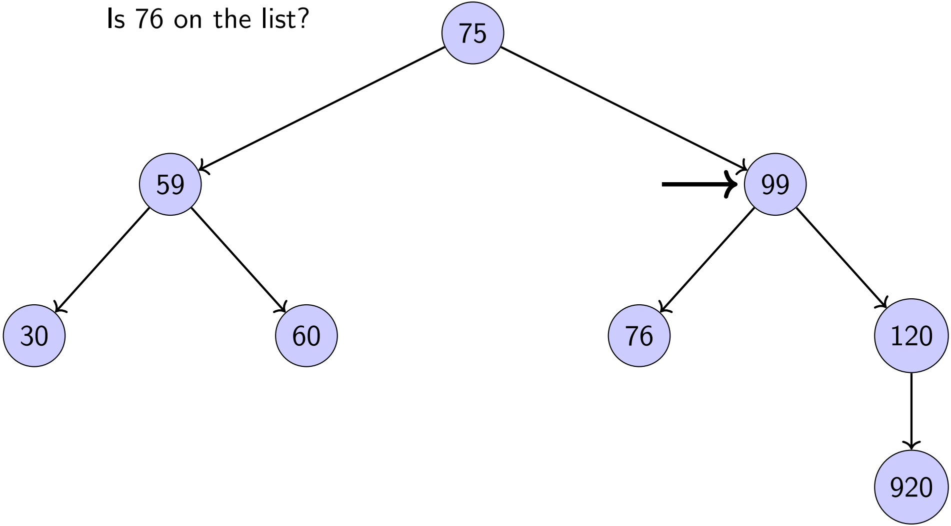
Search Tree

Is 76 on the list?



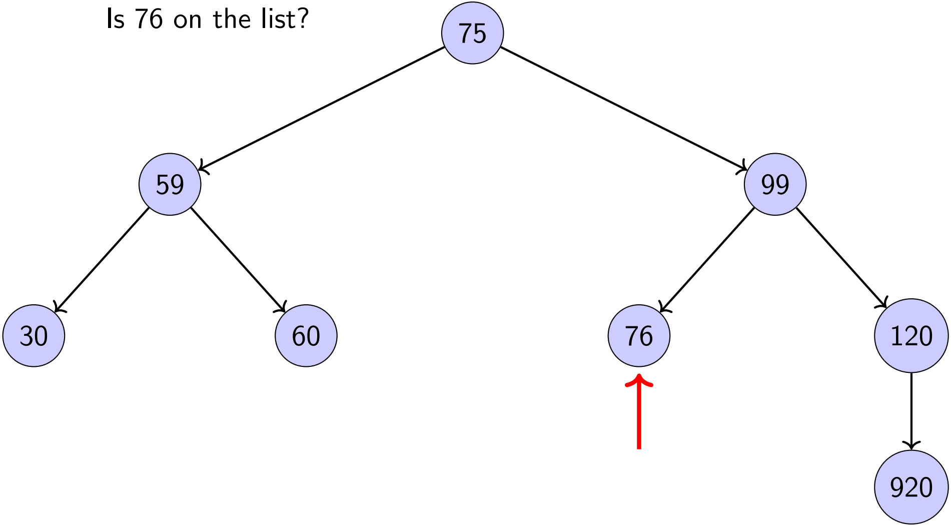
Search Tree

Is 76 on the list?



Search Tree

Is 76 on the list?



Search Tree

```python
def _get(self,key,currentNode):
    if not currentNode:
        return None
    elif currentNode.key == key:
        return currentNode
    elif key < currentNode.key:
        return
           self._get(key,currentNode.leftChild)
    else:
        return
           self._get(key,currentNode.rightChild)
```

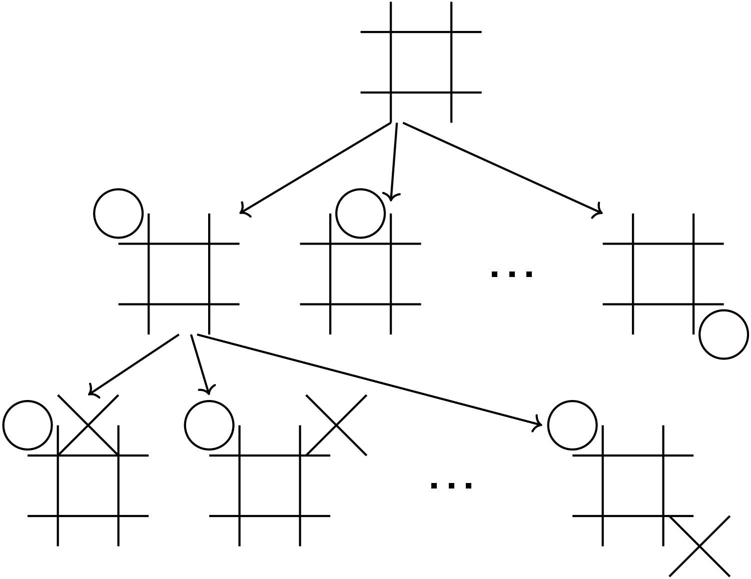
Search Tree

```
def _put(self,key,val,currentNode):
    if key < currentNode.key:
        if currentNode.hasLeftChild():
            self._put(key,val,currentNode.leftChild)
        else:
            currentNode.leftChild =
               TreeNode(key,val,parent=currentNode)
    else:
        if currentNode.hasRightChild():
            self._put(key,val,currentNode.rightChild)
        else:
            currentNode.rightChild =
               TreeNode(key,val,parent=currentNode)
```

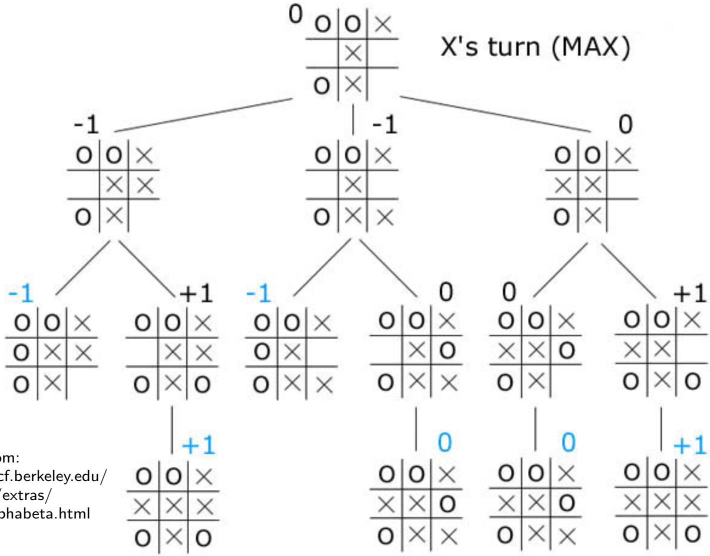
Search Tree

More Information:
http://interactivepython.org/runestone/static/pythonds/Trees/SearchTreeImplementation.html

Planning

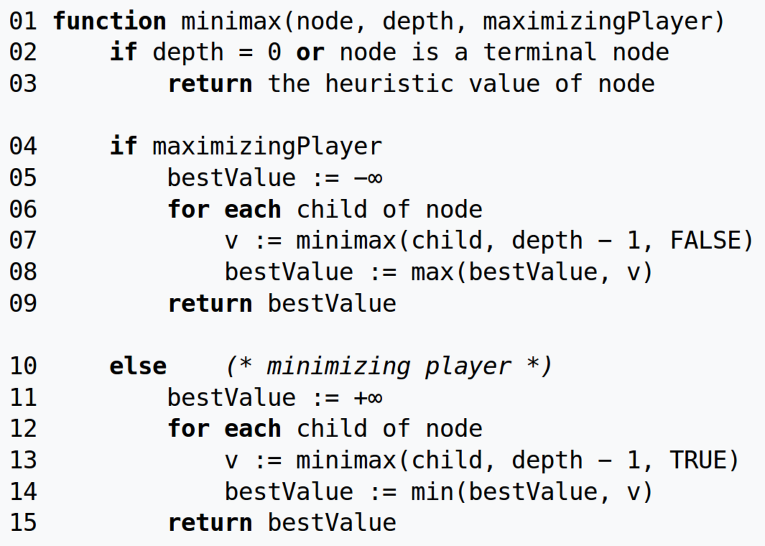


Planning – Minimax Search



From:
https://www.ocf.berkeley.edu/yosenl/extras/alphabeta/alphabeta.html

Minimax Search – Pseudocode



```
function minimax(node, depth, maximizingPlayer)
    if depth = 0 or node is a terminal node
        return the heuristic value of node

    if maximizingPlayer
        bestValue := −∞
        for each child of node
            v := minimax(child, depth − 1, FALSE)
            bestValue := max(bestValue, v)
        return bestValue

    else    (* minimizing player *)
        bestValue := +∞
        for each child of node
            v := minimax(child, depth − 1, TRUE)
            bestValue := min(bestValue, v)
        return bestValue
```

https://en.wikipedia.org/wiki/Minimax#Minimax_algorithm_with_alternate_moves

Outline

1. Private Members
2. Peer Feedback
3. DS Libraries
4. Problem Solving
5. Assignment
6. Conclusion

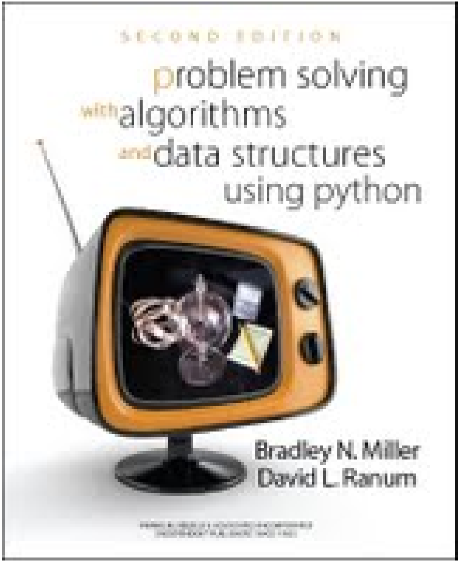
Assignment

1. Implement a logistic regression classifier, for a binary problem with two features. Use stochastic gradient descent for training. (3%)
2. Use the Monte Carlo method to approximate the value of π. (2%)

Outline

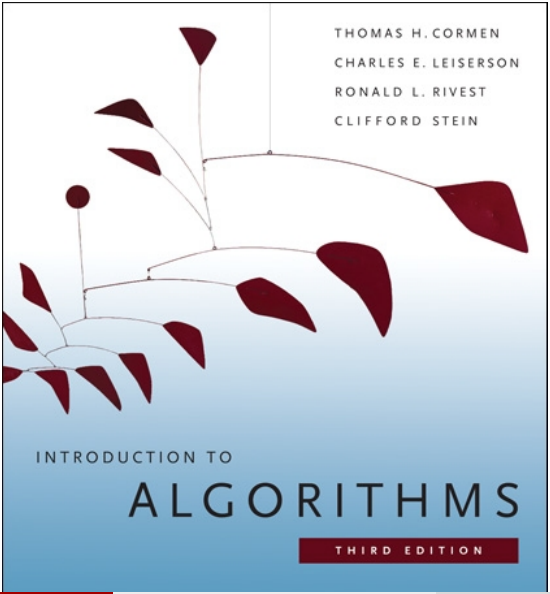
1. Private Members
2. Peer Feedback
3. DS Libraries
4. Problem Solving
5. Assignment
6. Conclusion

Keep Studying :)

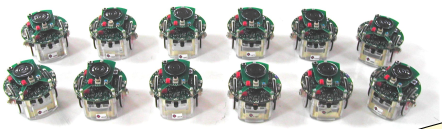

SECOND EDITION
problem solving with algorithms and data structures using python
Bradley N. Miller
David L. Ranum

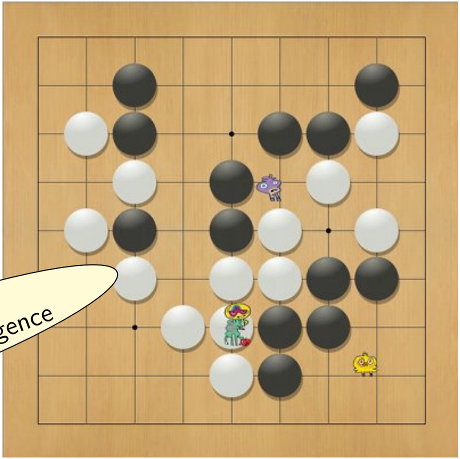
Problem Solving with Algorithms and Data Structures using Python
`http://interactivepython.org/runestone/static/pythonds/index.html`

Keep Studying :)



Advertisement...





AI research

SCC401-DS

Distributed Artificial Intelligence

http://www.lancaster.ac.uk/staff/sorianom/

l.marcolino@lancaster.ac.uk