index

October 24, 2021

ETE-2324: Data Structures and Algorithms

0.1 Course Contents

- **Introduction to Data Structures and Algorithms**
  - Reading:
    * PythonDS- Chapter 1
  - Notebook:Introduction
  - Lectures: Slides, PDF, HTML Latex
- **The Analysis of Algorithms**
  - Reading:
    * [Goodrich- Chapter 3]
    * PythonDS-Chapter 3
  - Notebook: Complexity Analysis
  - Lectures: Slides, PDF HTML Latex
  - Extra slides: CS161_at_Staford_Slides
- **Arrays**
  - Reading: [Goodrich- Chapter 5]
  - Lectures: Slides, PDF HTML Latex
- **Stack and Queue**
  - Reading: Stack – PythonDS- Chapter 4
  - Reading: Queue – PythonDS- Chapter 4
- **Searching**
  - Searching: Searching – PythonDS- Chapter 6
  - Sequential Search: PythonDS- Chapter 6.3
  - Binary Search: PythonDS- Chapter 6.4
  - Notebook: Searching

0.2 Additional Resources

- Data Stuctures and Algorithms Visulization – excellent resources for understanding both structures and algorithms.

0.3 Textbooks

- [**PythnDS**] Problem Solving with Algorithms and Data Structures using Python
- [**Goodrich**] Michael T. Goodrich, Roberto Tamassia, Michael H. Goldwasser. **Data Structures and Algorithms in Python** Wiley (2013)

0.4 Reference Books

- [**Cormen**] Cormen, Thomas, Charles Leiserson, Ronald Rivest, and Clifford Stein. **Introduction to Algorithms**. 3rd ed. MIT Press, 2009. ISBN: 9780262033848.

0.5 Environment Setup:

- Python 3 and Jupyter Installation - Python 3 Installation & Setup Guide - Anaconda Installation - Jupyter Installation Guide

0.6 Python Tutorials

- Part 1: Slides, Notebook, [HTML]python/(python_p1.html)
- Part 2: Slides,Notebook, HTML
- Part 3: Slides,Notebook, HTML
- List in Python: Notebook, HTML

```python
[3]: print(bool(0))
```

```
False
```

```python
[ ]: class Stack:

         def __init__(self):
             self.items = []

         def push(self, item):
             self.items.append(item)

         def pop(self):
             item = self.items[0]
             self.items[1:]
             return item

         def peek(self):
             return self.items[0]

         def size(self):
             return len(self.items)

         def is_empty(self):
             return self.items == []
```

```python
[5]: exp = "(a+b) / (3+4)"

     for symbol in exp:
         print(symbol)
```

```
(
a
```

```
+
b
)

/

(
3
+
4
)
```

[ ]: