**Lecture 31-10-21**

**The min in the lecture record**

11:00 references(pointers)

16:00 Assignment

18:30 id( )

33:00 copying list

37:33 Comparing two lists

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

1:08:00 Mutable vs immutable

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

1:36:30 shallow vs. deep copying

1:47:30 flip\_horizontal

**Question**

**Lecture 07-11-21**

**The min in the lecture record**

**MISSINGGGGGGGG**

**Lecture 6 14-11-21**

**The min in the lecture record**

11:36 Average case size

16:47 Constants Fade in importance

20:00 Binary Search Algorithm

21:00 Binary Average Search Algorithm

35:00 GCD algorithm

45:00 Running time of common python

Operation

1:11:44 Python Implements Lists

1:18:00 Dictionary Access and insert – O(1)

1:32:40 Algorithm with List

1:37:00 Sorting Algorithm

1:49:00 merge Sort Intuition

Question

סיבוכיות

O(של משהו)

**Week 7 Prepare Record Lecture 21-11-21**

**The min in the lecture record**

1:00 Recursion

2:53 examples- The king Arthur and Jhon Lennon

-if they have family member that is related to Jhon and

Related to me.

6:56 examples – N!

How much is 5!, we neem to check on 4! And then 3! And then 2!

And so on…

תמונה שמכילה טקסט

התיאור נוצר באופן אוטומטי

11:30 How to think Recursively?

* Problems that can be solving and describe by simple case
* What is the simplest case: The Base

14:40 We need that something will stop it in the simple case

16:00 Another example:

17:40 Recursion Vs loop

20:00 The call stack – How Recursion works?

**Week 7 Record Lecture 21-11-21**

**The min in the lecture record**

15:30 Fibonacci

23:00 An efficient version

36:00 Recap

39:00 Binary search

1:18:00 Quick Sort

1:46:00 The wort case, everts case

Questions-

1. Binary search: whey is log(n)???
2. Bubble sort

**Week 8 Prepare Record Lecture 28-11-21**

**The min in the lecture record**

Backtracking\_1\_sudoku

2:27 Brute Force, exponential

3:15 Example: Solve a Sudoku Puzzle

4:00 for the left upper number we will change to 9 option of of numbers (1,2,3,4,5,6,7,8,9) and have 9 easier problems

to solve

11:37 \_sudolu\_helper

Don’t understand the (1) base case-

Answer- Full board that we are printing

20:05 (2) if we already have number we will go for the next step

by recursion

(3) if now have number on the spot so we will try all the

Different options for it use of recursion

(4) if we didn’t succeed to find solution for options i try

We will delete the last step

24:00 A modification

Backtracking\_2\_All permutaions

2:00 helper

4:30 Base case : if the index is the last one we will print the lst

7:00 the backtracking: every time we try for all the option but

we are trying all the options we will go

for the lest case and start again.

Backtracking\_3\_Knight’s tour

**Week 8 Live Lecture 28-11-21**

Beginning def distribute n between 2 people

12:20 def distribute n between 3 people-loop

12:24 The complexity of distribute n between 3 people - loop

12:29 def distribute n between k people - recursion

-base case: add for the smallest distribution the coins for the first

13:03 Backtracking def distribute n between k people

-base case: when the index of distribution is equal k

13:25 Baktracking: Maze

-base case: if even to the starting point we can not go so return

-sec base case: if starting point is equal to end point

# \*\*\* Backtracking explain 67101 2021-11-28 שעור ב Time 1:10:30

**Week 9 first record Lecture W9p0**

14:30 Object Oriented Programming

20:00~ Examples

25:00~ Object Classes

- Objects Can Do Things

- Objects Can be Sent massage

30:00 The Basics of Objects

37:00 Each Object is an Instance of a **Class**

45:00 CONCLUTIONS-

Object Vocabulary

• Classes — Prototypes for objects

• Objects — Nouns, things in the world

• Constructor — For a Class, the way to create an

Object (that is, an Instance of the Class) and

initialize it

• Attributes — Properties an object has (sometimes

called "Data Attributes")

• Methods — Actions that an object can do

(sometimes called "Method Attributes")

• Messages — Communication from one object to

another, asking for a method to be used

Week 9 second record Lecture W9p1

2:20 How “Car” will be used

5:13 Class vs. Object

9:36 Initializing objects

15:40~ Methods

20~ Hiding the internal representation

25~ Ensuring internal consistency

30~ Application Programming Interfaces (APIs) and

design by Contract

– how programmers create and use its instances

32~ Suitcase Class

Week 9 third record Lecture W9p2

2:20 inheritance

5:00 python syntax

10:00 All things are objects

13:35 Object are mutable

23:00 Table of operators

27:00 Types and class

Week 10 prepare record Lecture

## Week10\_01\_LinkedLists

1:00 Linked lists

8:00 Inserting into a list

12:00 Removing from a list

16:00 Doubly-linked lists

## Week10\_02\_Stack

0:30 Stack ADT

10:00 A stack using a linked list

12:00 Improving the stack class

## Week 10\_03\_More Lists

00:30 Time Complexity

5:40 Recursion and Linked Lists

9:00 Another Method for Finding and Item in the list

## snail\_race

01:00 snail.py

05:00 snail\_race.py

17:00 race\_printer.py

Week 10 Live Lecture 12-12-2021

## First Hour

### Trees

14:06 General Graph

14:16 graphs Hard problems

14:20 3 coloring graph = sudoku problem

14:21 Definition Tree

14:25 Recursion and Trees

14:31 Trees that represent expressions

14:38 Computing the value of an expression

## Sec Hour

15:00 Polish Notation

15:06 Traversing a tree

15:11 Decision Trees

### Sorted Set- Data structure

15:25 Implementing data

15:31 Compression

### Binary Search Trees

Explaining

Comparing

Week 11 prepare record Lecture

## Week 11\_01\_Exceptions1

2:00 Handling the error

Try, except

8:00~ Several exception types

12:00~ Our own Error -> write own exception (**raise Exception**)

16:00~ The philosophy of exceptions

19:40 Definig new Exception class

22:00 Else and Finally(always do it)

## Week11b\_1\_dynamic\_parameters

0:30 varying number of parameters

2:00 Keywords and unknown keyword

04:00 Conclusion if args and kwargs

*Week11b\_2\_FunctionalProgramming*

0:30 introduction

05:00 lambda

08:55 function as a arguments

10:00 Type hints for function (from typing import callable)

*Week 10 Live Lecture 12-12-2021*

First Hour

*Currying*

*12:27 General – function(a,b,c) -> function (a)(b)(c)*

*12:33 examples-*

*Decorators*

*12:34 Example: print(“hi!”) >> \*\*\* hi! \*\*\**

*12:38 Example: Timing Functions*

*Memorization*

13:10 Idea: cach the results of comutation

Week 12 prepare record Lecture

## Week12-GUI-1-2

2:00 WHAT IS GUI?

3:00 The event loop

11:00 Widget

19:30 Buttons

Week 13 prepare record Lecture

## Week 13\_1\_Iterators

2:00 Rules of iterators

02:00 inter(list)

03:00 next(my\_iter, ”no more values”)

07:00 What loops actually do

10:00 why use iterators?

The size of amal and big iter are same!

## Week 13\_2\_Building\_Iterators

02:00 writing the class PrimesIter

05:00 itrertors in inters

## Week 13\_3\_Generators

0:30 yield

2:00 how to use Generators

08:30 building generator by list comprehension

09:30 Lazy evaluation