Turtle – draw a picture

Recursion, iterative drawing

Probably bounding box

Probably random colors

Probably random something

Written

Short answer

Mc

Writing examples

Greedy algorithms

Maing change with most coins of largest to smallest denomination

🡪one algorithm that chooses locally optimal paths

Has to make a decision, limited information at that point.

Doesn’t retracta decision

May or may not achieve a globallyt optimal solution

(optimality is change in lowest coins)is it optimal for 1,5,10,25 cents?)

Is it optimally generally?

* 3 comp. to problem soln

1. input – denomination of couins needed to change

2. make changes

Amount

List of available denominations

Sort out lists [smallest -> largest]

Reverse list

Make a copy so original isn’t destroyed

Store results in list of structures

3. print ou results

1. could read from text file

Alternatively prompt user for input:

SENTRNRL value – indicates that wewant to break from the loop

Class denomQuant():

\_\_slots\_\_ = )”(denom”,” quant”)

Struct = denomQuant()

Struct.denom = demon

Trar.quant = quaf

Def main():

Denomination list()

While True:

Val = int(inpt(“enter denom.(-1 to input).”))

If val ==-1:

Break

Else:

Denominations.append(val)

#denomination += val

Amt = omiq

2. inpt parameters into piltparaametieter\ ---- list of avaiavle domoalbysort the list = make it smail

Sort oout list [smallest to largest]

Class construct;

Define a new type of object/structure

Provide named componenets to thjat digit/structure

Class DenomQuant():

\_\_slots\_\_ = (“denom”, “quant”)

Looks like how we write functions (syntax)

Defins new structure (type)

Doesn’t create any instances

Can call denomQuant() to create an instance of it

Unintialized

May want to assign initial calies to denom nand quant

Range the list

Make a reply of list to test the original

Coins = amount % your curriculum

Amt = amt - - coins % curr\_down

Or

Amt = amounstoring out relii

New list representing number of froe=omm

Drawback is that our list cant be identicle even if coousese pyjon.ly

Alternatlyevelycold have list proceeed in pairs

Related list acan list or tuples as elements

Clarity – would like to know names for eaech data from

Maker function():

Struct = denomQuant()

Struct.denom = demon

Struct.quant = quant

Return struct

#create an instance of our structure

#initialize it

#return it

Def majerDenomQuant(denom, quant)

Complexity:

1. input – n values when done

2. make change –

Make copy of list (N)

Sorting list (N^2)

Reverse list (N)

Loop – iterates at most (N times

Total loop is order of N

Total nelgjtlonu

2. display

Constant amount of work N

N+N2+N+N+N

Largest one dominates, N^2