

Name: _____

andrewID: _____

Practice Final Exam - Instructions:

- Do not open this exam until the instructor tells you to.
- This final exam tests material from the entire course, with an emphasis on material from weeks 8-14. You have **3 hours** to take the exam.
- If you have a clarification question, raise your hand and a proctor will come help you.
- You must complete the exam **individually**. You may refer to paper notes during the exam, but do not communicate with anyone else.
- Show your work to get full credit.
- Make sure to display values **unambiguously**. Strings should be in quotes (unless they're being printed), floats should have decimals, Booleans should be capitalized.

1. Code Writing - Code Fundamentals [6pts]	2
2. Code Reading - Loops [7pts]	3
3. Code Reading - Dictionaries [6pts]	4
4. Code Writing - Trees [6pts]	5
5. Free Response - Parallel Programming [6pts]	6
6. Free Response - Internet and Security [7pts]	7
7. Code Reading - Managing Large Code Projects [6pts]	8
8. Code Writing - Data Analysis [7pts]	10
9. Code Writing - Simulation [6pts]	12
10. Free Response - Machine Learning [6pts]	14
11. Short Answer - Functions [4pts]	15
12. Short Answer - For Loops [3pts]	15
13. Short Answer - Efficiency [3pts]	16
14. Short Answer - Graphs [4pts]	17
15. Short Answer - Internet [4pts]	18
16. Short Answer - Simulation [4pts]	19
17. Short Answer - Machine Learning [3pts]	20
18. Short Answer - CS History [4pts]	21
19. Short Answer - CS Ethics [4pts]	21
20. Short Answer - CS Future [4pts]	22

1. Code Writing - Code Fundamentals [6pts]

Write a function `waterForm(temp)` which takes a floating point number, `temp`, and returns a string, either "ice", "water", or "steam". The function takes a temperature in degrees Fahrenheit, converts it to degrees Celsius, and then returns the form water will take at that temperature. You should solve the problem by following the given algorithm:

1. Convert the temperature from Fahrenheit to Celsius using the formula:
 $^{\circ}\text{C} = (^{\circ}\text{F} - 32) \div 1.8$. Store this new temperature in a variable.
2. If the temperature in $^{\circ}\text{C}$ is less than or equal to 0, return "ice".
3. If the temperature in $^{\circ}\text{C}$ is in the range (0, 100], return "water".
4. If the temperature in $^{\circ}\text{C}$ is greater than 100, return "steam".

For example, `waterForm(105.2)` would convert 105.2 -> 40.7, then return "water".

```
def waterForm(temp):  
    c = (temp - 32) / 1.8  
    if c <= 0:  
        return "ice"  
    elif c <= 100:  
        return "water"  
    else:  
        return "steam"
```

2. Code Reading - Loops [7pts]

Consider the following Python program. What will be printed when the whole program executes? Write your answer on the right.

<pre>def fn1(x, y): print("In fn1:", x, ":", y) total = 0 while x > y: total = x + total x = x - 2 print("Loop:", x, ":", y) return total def fn2(x): print("In fn2:", x) z = fn1(x, x - 5) print("z:", z) return z + x print("Begin") print("Result:", fn2(9)) print("End")</pre>	<pre>Begin In fn2: 9 In fn1: 9 : 4 Loop: 7 : 4 Loop: 5 : 4 Loop: 3 : 4 z: 21 Result: 30 End</pre>
---	---

3. Code Reading - Dictionaries [6pts]

Consider the following code:

```
def mystery(wordList):  
    d = { }  
    for word in wordList:  
        length = len(word)  
        if length not in d:  
            d[length] = [ ]  
        if word not in d[length]:  
            d[length].append(word)  
    return d
```

What will be returned if we call `mystery(["zoo"])`?

```
{ 3 : [ "zoo" ] }
```

What will be returned if we call `mystery(["zoo", "sky", "train", "sky"])`?

```
{ 3 : [ "zoo", "sky" ], 5 : [ "train" ] }
```

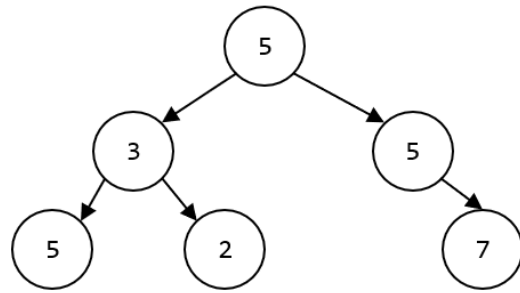
Is it possible to call `mystery` on a (non-empty) input that results in the function returning an empty dictionary? If yes, provide an example input. If no, explain why not.

No, because the first value in the list will always be added to the dictionary

4. Code Writing - Trees [6pts]

In the box below, write the function `countValue(tree, value)` that takes a tree in our dict structure and a given value as parameters and returns how many times the given value appears in the tree. For example, if you call the function `countValue` with the tree below and the value 5, it would return 3 as there are 3 nodes with value 5 in the tree.

```
tree = {
  "contents": 5,
  "left": {
    "contents": 3,
    "left": {
      "contents": 5,
      "left": None,
      "right": None },
    "right": {
      "contents": 2,
      "left": None,
      "right": None } },
  "right": {
    "contents": 5,
    "left": None,
    "right": {
      "contents": 7,
      "left": None,
      "right": None } } }
```



`countValue(tree, 5) → returns 3`

```
def countValue(tree, value):
    if tree == None:
        return 0
    else:
        result = 0
        result += countValue(tree["left"], value)
        result += countValue(tree["right"], value)
        if tree["contents"] == value:
            return result + 1
        else:
            return result
```

5. Free Response - Parallel Programming [6pts]

Wrapping a present takes four steps:

1. **B**: Put the present in a box
2. **P**: Wrap the box in gift wrap paper
3. **R**: Tie a ribbon around the wrapped box
4. **N**: Attach a name tag to the ribbon

Steps **B**, **R**, and **N** each take only one minute, but step **P** takes three minutes. You can indicate this by following the initial P with dashes (-).

Assume you have four people who are all working together to wrap presents. First, fill in the table below to design a **pipelining** approach for wrapping presents across four workers. Note that each column represents a single minute of time.

	1	2	3	4	5	6	7	8	9	10
A	B	B	B	B	B	B	B	B	B	B
B		P	-	-	P	-	-	P	-	-
C					R			R		
D						N			N	

How many gifts can you wrap in 10 minutes with pipelining ?	2
--	----------

Now fill in the next table to design a **parallel** approach similar to MapReduce for wrapping presents across four workers. Each column represents a minute of time.

	1	2	3	4	5	6	7	8	9	10
A	B	P	-	-	R	N	B	P	-	-
B	B	P	-	-	R	N	B	P	-	-
C	B	P	-	-	R	N	B	P	-	-
D	B	P	-	-	R	N	B	P	-	-

How many gifts can you wrap in 10 minutes with the parallel approach?	4
--	----------

6. Free Response - Internet and Security [7pts]

Ben wants to send a private message to Wei over the internet using asymmetric encryption. A third person, Sol, wants to read the message without either person's permission.

Fill in the blanks below to describe the process used to send this message securely. Each blank should use one of the words in the boxes below. Some words may be used multiple times.

break browser certificate ciphertext	decode DNS server encode IP address	ISP packet password plaintext	private key protocol public key router
---	--	--	---

1. Ben looks up Wei's public key on the internet and uses it to encode his message.
2. Ben sends the ciphertext through the internet using lots of small packets.
3. While the message travels through the internet, it passes through many routers.
4. Sol intercepts the message while it's traveling through the internet and attempts to break the code without knowing the private key.
5. Wei receives the message. She uses a private key to decode it.

7. Code Reading - Managing Large Code Projects [6pts]

We want to write Python code to implement a text-based game of Connect Four. If you've never played this game before, all you need to know is that it is a two-player game where players take turns putting pieces on a two-dimensional grid, and whoever gets four pieces in a row first wins the game.

Assume we already wrote four helper functions for this game:

- `makeBoard()`, which takes no input and returns a 2D list representation of the board
- `displayBoard(board)`, which takes the 2D list representation of the board and displays it nicely in the interpreter
- `makeMove(board, player)`, which takes the 2D list representation of the board and the current player (either "red" or "blue"), gets a valid move from the player, applies that move to the board, and returns the updated board
- `isGameOver(board)`, which takes the 2D list representation of the board and returns True if the game is over and False otherwise.

We then wrote the main game function as follows:

```
1 def connectFour():
2     b = makeBoard()
3     displayBoard(b)
4     currentPlayer = "red"
5     while isGameOver(b) == False:
6         b = makeMove(b, currentPlayer)
7         displayBoard(b)
8
9         if currentPlayer == "red":
10             currentPlayer = "blue"
11         else:
12             currentPlayer = "red"
```

Continued on the next page

Continued from the previous page

Read over the description of the helper functions and the code to get a sense of how they work, then answer the following questions. Your answers should be in **general terms** - giving a high-level explanation of what's going on, not repeating the code itself.

What is the purpose of the variable `b` used throughout the code?

Holds the current state of the board

When `displayBoard` is called on line 7 in the first iteration of the loop, what will it display?

The board after the first move is made

How does the code on lines 9-12 relate to the way the game works?

Alternates between players

When will the loop stop running?

When the game is over

8. Code Writing - Data Analysis [7pts]

You have data about some books you have bought. The data is a 2D list and includes the title of the book, the genre, the number of pages and the price of the book, all as **strings**. A sample is shown below:

```
books = [  
    ["title", "genre", "pages", "price"],  
    ["The Hobbit", "fantasy", "310 pages", "$12.99"],  
    ["1984", "dystopian", "328 pages", "$15.50"],  
    ["Dune", "sci-fi", "688 pages", "$11.99"],  
    ["The Name of the Wind", "fantasy", "662 pages", "$16.99"],  
    ["Neuromancer", "sci-fi", "271 pages", "$18.75"],  
    ["Mistborn", "fantasy", "295 pages", "$11.25"]  
]
```

Write the function **findBooks(*books*, *genre*, *pages*, *price*)** which takes a 2D list, a string of a genre, an integer for number of pages and a float for price. The function returns a list of book titles in that genre with a number of pages less than the given pages value and a price less than the given price value. For example:

```
findBooks(books, "fantasy", 350, 13.00) → ["The Hobbit", "Mistborn"]  
# as we have bolded in the dictionary above
```

```
findBooks(books, "sci-fi", 300, 15.00) → [ ]  
# no sci-fi books < 300 pages and < $15.00 in the dictionary
```

Hint: you'll need to do some data processing. Think about how to skip the header and how to turn the number of pages and price into numbers. **The data will always be formatted as shown above**, though different entries may be used after the header.

There is space to write your answer on the next page.

Please write your answer to the problem on the previous page below.

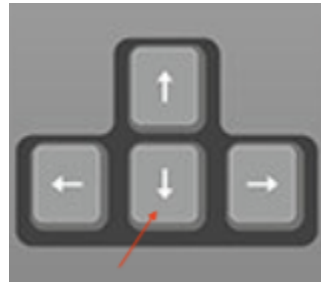
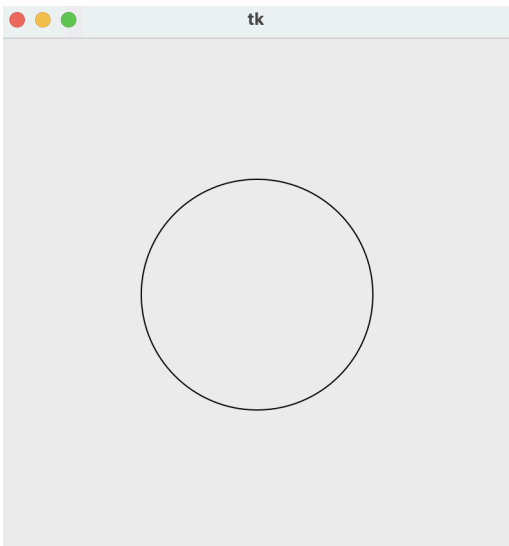
```
def findBooks(books, genre, pages, price):
    result = [ ]
    for entry in books:
        if entry[1] == genre:
            bookPages = int(entry[2].split(" ")[0])
            bookPrice = float(entry[3][1:])
            if bookPages < pages and bookPrice < price:
                result.append(entry[0])
    return result
```

9. Code Writing - Simulation [6pts]

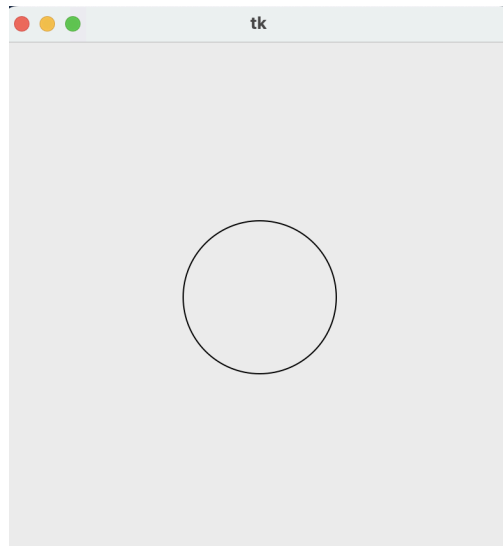
You want to create an event-based simulation using our simulation framework where a circle grows or shrinks based on keyboard input. The circle is drawn in the center of the 400 x 400 screen with an initial radius of 100. The radius increases by 10 pixels when you press the "Up" key and decreases by 10 pixels when you press the "Down" key. You can see an example of the simulation below.



When you press the up key the circle size increases by 10.



When you press the down key the circle size decreases by 10.



Fill in the functions below to implement the described simulation. **Note:** you can assume the starter code for the simulation framework has already been included.

Problem continues on next page

Problem continues from previous page

```
def makeModel(data):  
    data["r"] = 100
```

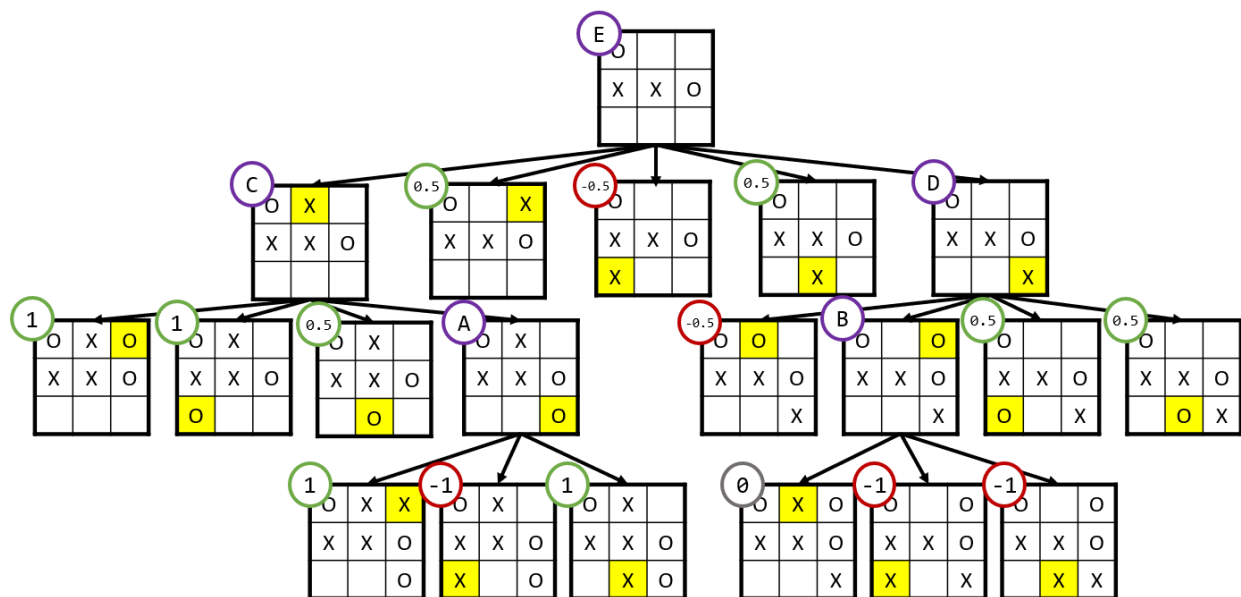
```
# canvas.create_oval(left, top, right, bottom) draws an oval  
  
def makeView(data, canvas):  
    canvas.create_oval(200 - data["r"], 200 - data["r"],  
                      200 + data["r"], 200 + data["r"])
```

```
# event.keysym will hold "Up" when the user clicked the up key and  
# "Down" when they clicked the down key  
  
def keyPressed(data, event):  
    if event.keysym == "Up":  
        data["r"] += 10  
    elif event.keysym == "Down":  
        data["r"] -= 10
```

10. Free Response - Machine Learning [6pts]

We have built a partial game tree for the game Tic-Tac-Toe, where the AI is playing X, the user is playing O, and the AI has the next move. Some states have been propagated down several levels; others have an estimated score using a heuristic.

Using the **minimax** algorithm described in class, calculate the scores for the states that have been labeled with letters, then select **all** the next moves that the AI might select if it follows the algorithm to its natural conclusion.



Game state	A	B	C	D	E
Score	1	0	0.5	-0.5	0.5

Which move(s) might the AI choose to take next based on the minimax results?

- ☒ Place X in top-middle
- ☒ Place X in top-right
- ☐ Place X in bottom-left
- ☒ Place X in bottom-middle
- ☐ Place X in bottom-right

11. Short Answer - Functions [4pts]

For each of the following function calls (where the relevant function call is bolded), write its **argument(s)** and its **returned value(s)**.

Function Call	Argument(s)	Returned Value(s)
result = pow (5, 2)	5, 2	25
name = "Sue" print ("Name: " + name)	"Name: Sue"	None

12. Short Answer - For Loops [3pts]

Fill in the blanks for the range expression so that the code prints every fifth number from 10 to 100, including those two numbers. (So it should print 10, 15, 20, 25... 95, 100).

```
for i in range(____10____, ____101____, ____5____):  
    print(i)
```

13. Short Answer - Efficiency [3pts]

Describe a generic input for the following function that would result in **best-case efficiency**, then describe a generic input that would result in **worst-case efficiency**. This generic input must work at **any possible size**.

```
def findHello(lst):  
    for el in lst:  
        if el == "hello":  
            return True  
    return False
```

What is a generic **best case input** for findHello?

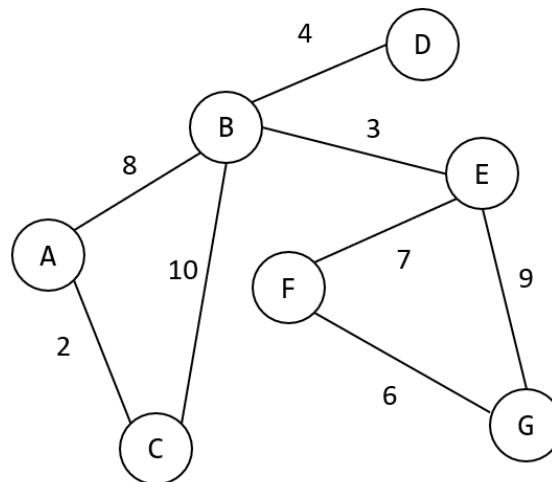
The first string in the list is "hello"

What is a generic **worst case input** for findHello?

The list does not contain "hello" / the last string in the list is "hello"

14. Short Answer - Graphs [4pts]

Given the graph below, identify the following:



How many nodes does the graph have?	7
How many edges does the graph have?	8
What are the neighbors of node E?	B, F, G

Select **all** of the following descriptors that are true for the above graph.

- ☐ Directed
- ☒ **Undirected**
- ☒ **Weighted**
- ☐ Unweighted

15. Short Answer - Internet [4pts]

Which of the following are true about how **packets travel across the internet**? Select all that apply.

- ☐ The sender specifies exactly which routers a packet must visit
- ☒ **The internet figures out the route of the packets based on the destination IP address**
- ☐ All packets from the same webpage always take the same path
- ☒ **Different packets can take different routes to reach the same destination**
- ☐ All packets from the same webpage always arrive in order
- ☐ All packets sent by a webpage are guaranteed to arrive at the destination
- ☐ All packets must use the same router when sent from receiver back to the sender
- ☐ A full website travels as one individual packet in the internet

Which of the situations below demonstrate examples of **fault tolerance** on the internet?

- ☒ **If a packet is missing, the browser can request it again**
- ☐ If a packet has been corrupted, the browser cannot detect the corruption
- ☐ If your DNS request fails, you cannot get the IP address of a website
- ☐ If your ISP goes down, you can still connect to the internet via other routers
- ☒ **If a router fails, the traffic is rerouted to other routers**
- ☐ If a company's website goes down, the internet shuts down the website permanently

16. Short Answer - Simulation [4pts]

You want to use Monte Carlo methods to determine how likely it is that when flipping a set of 3 coins in the air the side will be the same (i.e., all 3 coins end up heads or all 3 coins end up tails).

You've already written the method `flipCoins()`, which generates coin flips for all 3 coins and returns a string holding the three results, "H" for heads and "T" for tails. For example, flipping Heads, then Tails, then Heads would return the string "HTH".

Fill in the blanks in the function `getExpectedValue()` to determine the odds that 3 coins are all flipped to the same side.

(Note that this function does not take a parameter. This was done on purpose.)

```
def getExpectedValue():
```

```
    total = 0
```

```
    for trial in range(_____1000_____):
```

```
        result = _____flipCoins()_____
```

```
        if _____result == "HHH" or result == "TTT"_____:
```

```
            total = total + 1
```

```
    return _____total / 1000_____
```

17. Short Answer - Machine Learning [3pts]

What combination of **learning category** and **reasoning category** would you use for each of the following scenarios?

Train a robot to pick up fragile objects like eggs by continuously adjusting its grip strength and finger position while monitoring pressure.

Learning category you choose: ☐ Supervised Learning ☐ Unsupervised Learning ☒ Reinforcement Learning ☐ No learning	Reasoning category you choose: ☐ Classification ☐ Regression ☐ Clustering ☒ No reasoning
--	--

Estimate how many inches of snow (1", 3", etc) we will get this week based on historical weather data from Pittsburgh.

Learning category you choose: ☒ Supervised Learning ☐ Unsupervised Learning ☐ Reinforcement Learning ☐ No learning	Reasoning category you choose: ☐ Classification ☒ Regression ☐ Clustering ☐ No reasoning
--	--

Investigate potential groupings of people based on their favorite movie genre.

Learning category you choose: ☐ Supervised Learning ☒ Unsupervised Learning ☐ Reinforcement Learning ☐ No learning	Reasoning category you choose: ☐ Classification ☐ Regression ☒ Clustering ☐ No reasoning
--	--

18. Short Answer - CS History [4pts]

Which of the following is **not** a correct statement about a factor that contributed to the first concepts of how computers would work?

- ☒ **The Analytical Engine was the first computer that was constructed**
- ☐ The first program was written to calculate Bernoulli numbers using a series of states
- ☐ The Church-Turing Thesis gave a broad definition of what could be done on a computer
- ☐ All modern classical computers can be simplified to a Turing Machine

Which of the following is **not** a correct statement about a factor that contributed to computers becoming personal devices?

- ☐ Microprocessors made it possible to put computers in all sorts of devices, like clocks and calculators
- ☐ Transistors made computers significantly smaller, since they were so much smaller than vacuum tubes
- ☒ **Companies like Microsoft and Apple who attended the Mother of All Demos immediately built the first GUIs in their new personal computers**
- ☐ The Mother of All Demos introduced many exciting new technologies, including the mouse, the graphical user interface (GUI), and video conferencing

19. Short Answer - CS Ethics [4pts]

Which one of the following is **not** a concern with the notice-and-choice model of data collection that we discussed in class?

- ☐ Reading privacy policies takes too much time
- ☒ **Privacy policies are inaccurate**
- ☐ Consumers don't have realistic alternatives when they disagree with privacy policies

Which one of the following is **not** a concern with automated decision making that we discussed in class?

- ☐ Humans who are supposed to supervise algorithms may trust the algorithm's output without questioning it
- ☐ It may not be clear who to hold accountable for mistakes made by the algorithm
- ☐ The algorithm may reinforce existing societal biases
- ☒ **Humans may forget how to make the decisions themselves after relying on the algorithm for too long**

20. Short Answer - CS Future [4pts]

Which of these **best** describes the way a GAN can generate realistic images?

- ☐ Two models pass an image back and forth, each making adjustments until the quality of the image is high enough.
- ☐ The model identifies patterns that exist in real images and chooses a subset of patterns to include in the generated image
- ☐ The model finds real images online and slightly changes them so they're different from before
- ☒ **Two models compete with each other, one trying to tell real images from fake and the other trying to generate good images**

Which of these **best** describes how quantum algorithms are different from regular algorithms?

- ☒ **They use quantum bits instead of regular bits**
- ☐ They use a quantum programming language instead of a regular programming language
- ☐ They use quantum data structures instead of regular data structures
- ☐ They only work in theory, whereas regular algorithms can actually be implemented and executed with code