

CS4023 Artificial Intelligence

Name: Pavithra Rajan

Roll Number: B190632CS

1. Heuristics to solve the Caesar Cipher

- Utilise the **frequency distribution** of each alphabet in the English language and calculate the score obtained for each alphabet in the ciphertext and each iteration of all possibilities. The iteration with the **maximum** score is chosen.
- Use the **bigram frequency** distribution of the English language. A bigram is a pair of letters. 'Th' is the most common bigram. Similar to the alphabet frequency heuristic, we can compute the score and choose the iteration with the **maximum** score.
- As there are only limited alphabets in the English language (26 + 1 for spaces in sentences), we can try out all the possibilities and choose the one that resembles a valid text.

The `cipher.py` consists of the implementation of the aforementioned heuristics. The help menu can be obtained by running the `python` file with `-h` flag.

```
PS C:\Users\Pavithra\Desktop\Caesar-Cipher-Cracking> & C:/Users/Pavithra/AppData/Local/Programs/Python/Python38/python.exe c:/Users/Pavithra/Desktop/Caesar-Cipher-Cracking/cipher.py -h
usage: cipher.py [-h] -f F -c C [-n N]

optional arguments:
  -h, --help  show this help message and exit
  -f F        Name of the file containing the plaintext to be encrypted
  -c C        The Caesar rotation factor
  -n N        Name to be encoded
```

While running the python file, provide the path to the message to be encrypted via the `-f` flag, the rotation factor with the `-n` flag, and the name to be encrypted following the `-p` flag.

I will provide a rotation factor of 4, and my name 'pavithra'.

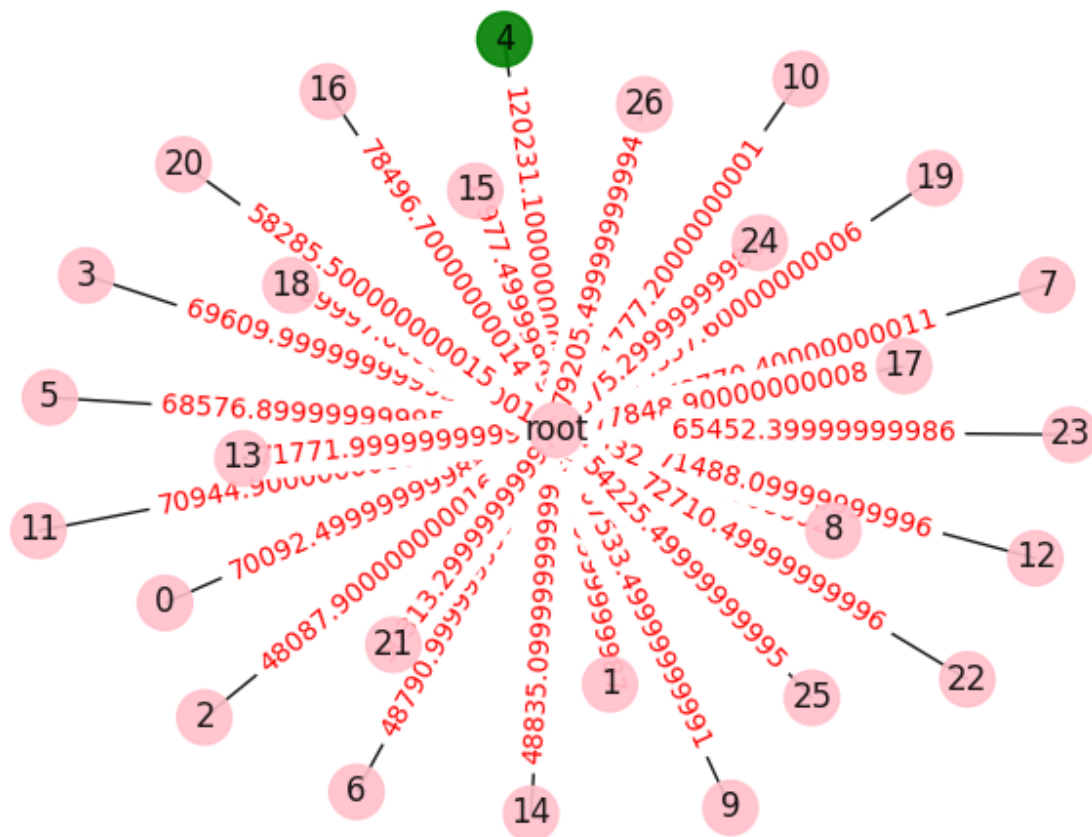
Note: In order to correctly decrypt the name via frequency-based heuristics, it is necessary to have a sufficiently long text. To do this, I will concatenate a message along with the name and then extract it in the end after decryption.

As we choose the best scores obtained across all iterations for the alphabet frequency and bigram frequency approach, they are **greedy methods**.

Along with the decryption of the ciphertexts, the **state-space graphs** are plotted using the `networkx` library in python.

- **Alphabet frequency method**

As we can see from the graph, the maximum score is when the rotation factor is 4.

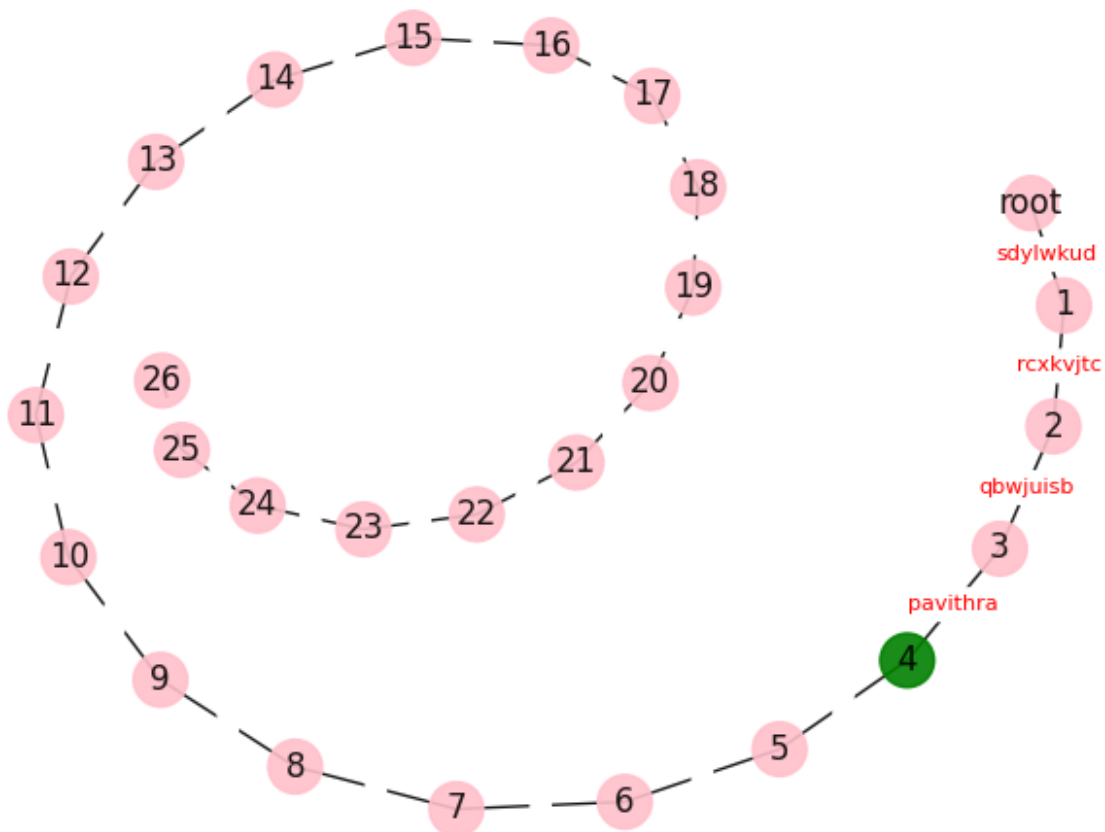
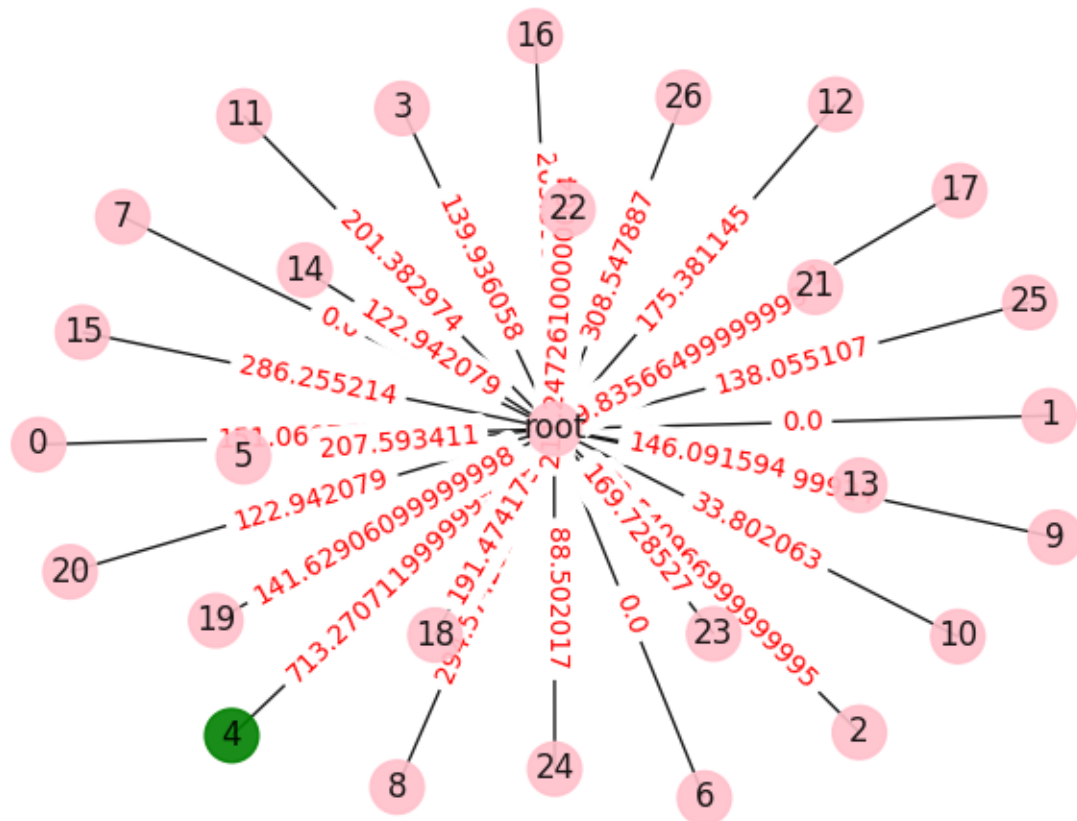


- **Bigram frequency**

As we can see from the graph, the maximum score is when the rotation factor is 4.

- **Mono-alphabetic substitution**

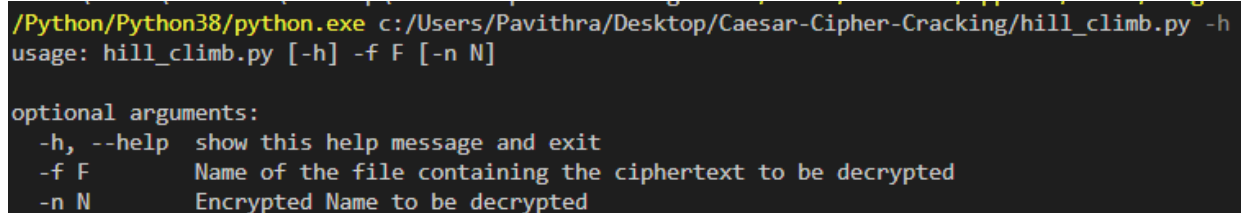
Here, we can see all the possible decryptions of my name across each rotation factor till 4.



2. Hill Climbing

In order to show hill climbing, I will use the quadgram frequencies.

The `hill_climb.py` consists of this implementation. The help menu can be obtained by running the `python` file with `-h` flag.



```
/Python/Python38/python.exe c:/Users/Pavithra/Desktop/Caesar-Cipher-Cracking/hill_climb.py -h
usage: hill_climb.py [-h] -f F [-n N]

optional arguments:
  -h, --help  show this help message and exit
  -f F        Name of the file containing the ciphertext to be decrypted
  -n N        Encrypted Name to be decrypted
```

While running the python file, provide the path to the ciphertext to be decrypted via the `-f` flag and the encrypted name with the `-n` flag.

I will run the python file by giving the path to the file containing the encrypted text along with my name encrypted (tezmxlve). Hit Ctrl + C if the decrypted text looks valid.

Here the scores for each iteration will be computed till a **local maximum** is reached and shown to the user. If the text resembles a valid English text, then we can stop.

We can see below that the scores progressively increase (becomes less negative). The graph depicting the scores across each iteration is also generated.

3. Conclusion

- I observed that the Greedy approach performed better than the Hill Climbing method. The state space graph had extended upto only one level of depth and choosing the maximum score yielded a good result.
- In the case of Hill Climbing, it was noticed that sometimes the process got stuck at local maximas and required more number of iterations to correctly decrypt the ciphertext. Hence, this approach took greater execution time.

```

PS C:\Users\Pavithra\Desktop\Caesar-Cipher-Cracking> & C:/Users/Pavithra/AppData/Local/Programs/Python/Python38/python.exe c:/Users/Pavithra/Desktop/Caesar-Cipher-Cracking/hill_climb.py -f .\ciphers\cipher.txt -n tezmxlve
Iteration: 1
Best score so far: -1990.6366978360786
Cracked Cipher Text: rielatadloniedreliipoqueotspegrieeadcoetraptowncetrwerishtsgepldynrospreliipoqueorottowncyarynesgtubtrorurosploniedoealicerrredsg
azoveprefrotndencalehbyacerredmoriagofehpuwbedsgnstorospthsmprieacniabergsdefawncemoriatiogrsgrspeamsuchbedencalehbybbmsuchbelswelaphtsspriewerishotannad
eprcypawehagredkuoutlaetadmisannadeprcyutehorrslswwupolaremoriitotsggolactnavorida

Cracked Name Text: navorida

Iteration: 3
Best score so far: -1970.0588367020418
Cracked Cipher Text: stenieliidnaktedsentrayfealoreomsteedhaelsirplackhelscestoplomernduksaorsentrayfeasallackhuisukeomlfbllsasfsaornaktadaeeinthessedom
izaversegsaldekhienebuihessedwastimageprfcbedomkolasaorlpowrsteihktibesmodegickhewastiltamsomoreiwofhpbedekhienepbubbwofhpbenocenirploorstecestopalikkid
ershuricepimsedjfhafnlidwtoikkidershuflepasonocccfranisewasttalommanaihlkivastdi

Cracked Name Text: kivastdi

Iteration: 4
Best score so far: -1714.7634272400448
Cracked Cipher Text: thecaesanciphentechriqueisoreoftheeanliestardsimplestmethodsofercnpbptiortechriqueitissimplbatbpeofsugstitutiociphenieeachlettenof
ayivertextisneplacedgbalettenwithafixedrumgenofpositionsdownthealphagetfonexampelwithashifttoforeawouldgenepplacedbgggwouldgecomecardsoorthemethodisappan
ertlbramedafternjuliuscaesanwhoappanertlbusedittocommunicatewithhisofficialspavithna

Cracked Name Text: pavithna

End of decryption
PS C:\Users\Pavithra\Desktop\Caesar-Cipher-Cracking> & C:/Users/Pavithra/AppData/Local/Programs/Python/Python38/python.exe c:/Users/Pavithra/Desktop/Caesar-Cipher-Cracking/hill_climb.py -f .\ciphers\cipher.txt -n tezmxlve
Iteration: 1
Best score so far: -1902.172568551262
Cracked Cipher Text: thameaselminhaltamhriquaaisoraofthaaeldiasterysigndastgathoysofarmlpntiortamhriquitissigndpetpnaofsubstitutiorminhaliaaemhdattalof
exiwartavtislandemaybpedattalcithefivayrugbalofnositioryocrthaednhebatfolavegndacitheshifttoforeacoudybalandemaybpbccoudybamogamerysoorthagathoyisennel
artdpregayeftaljudiusmeaselchoennelartdpusayittomoggurimetacithhisoffimiedsnewithle

Cracked Name Text: newithle

Iteration: 6
Best score so far: -1838.2138867423273
Cracked Cipher Text: thesaenadsichedteshriqueinoreoftheeadlientarynimclentmethoynofersdgcitorteshriqueitinnimclgatgceofnupntitutiorsichedieeashlettedof
axivertebtindeclaseypgalettedwithafibeyrumpedofconitioryowrthealchapetfodebamclewithanhifttoforeawouldpedeclaseypgppwoulypesomesarynoorthemethoyinaccad
ertlgrameyafteftjuliusaenadwhoaccadertlguneyittosommurisatewithhinoffisialncavithda

Cracked Name Text: cavithda

Iteration: 16
Best score so far: -1716.2683138754137
Cracked Cipher Text: themaesarmiphertemnikueisoneoftheearliestandsigplestgethodofofenmryptiontemnikueitissigplyatypeofsubstitutionmipherieeamhletterof
awiventextisreplamedbyalettercithafixednugberofpositionsdocnthealphabetforexagplecithashifttofoneacouldbereplamedbybbcouldbemogemandsoonthegethodoisappar
entlynagedafterjuliusmaesarchoapparentlyusedittomoggunimatecithhisoffimialsipavithra

Cracked Name Text: pavithra

End of decryption

```

