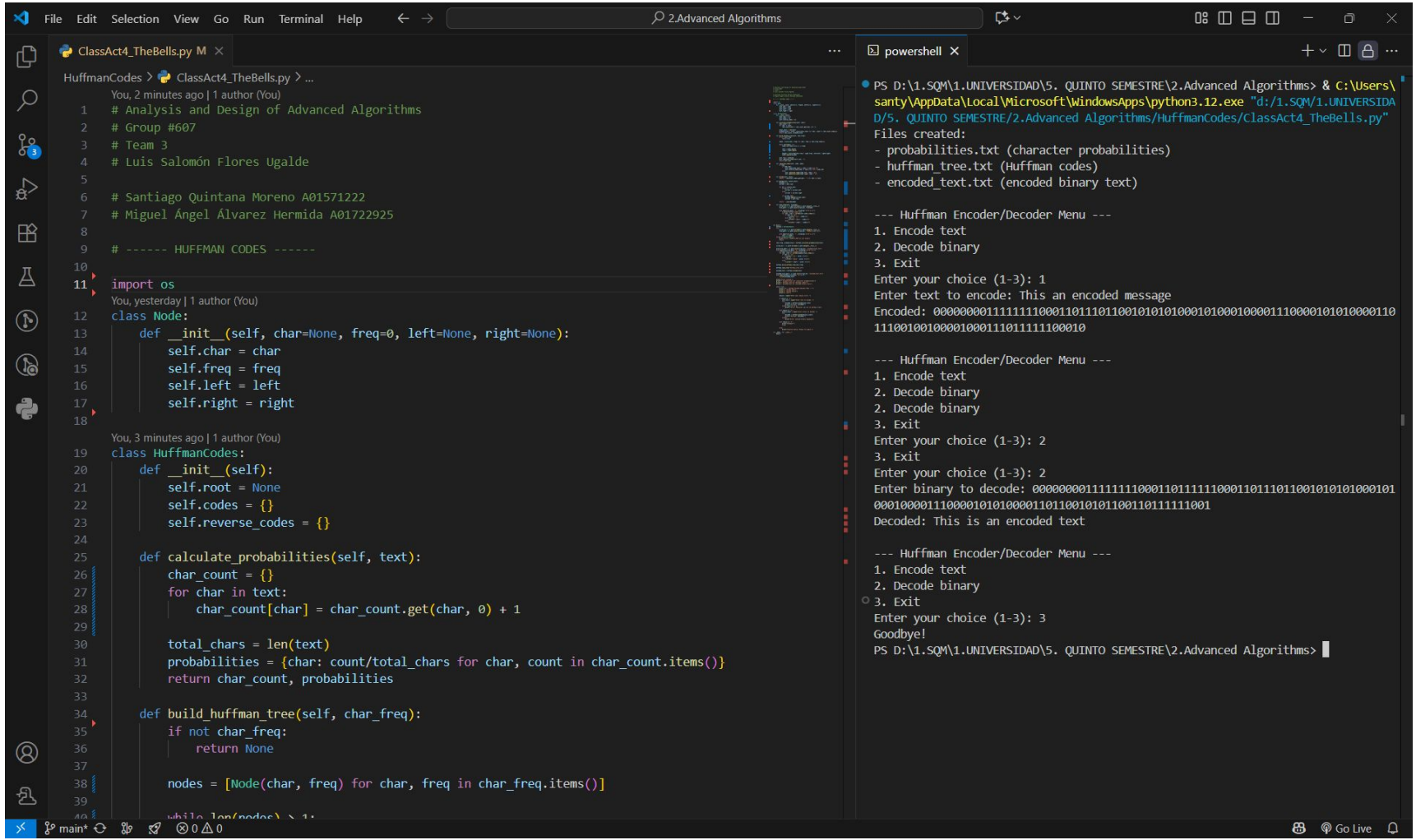
Tecnológico de Monterrey

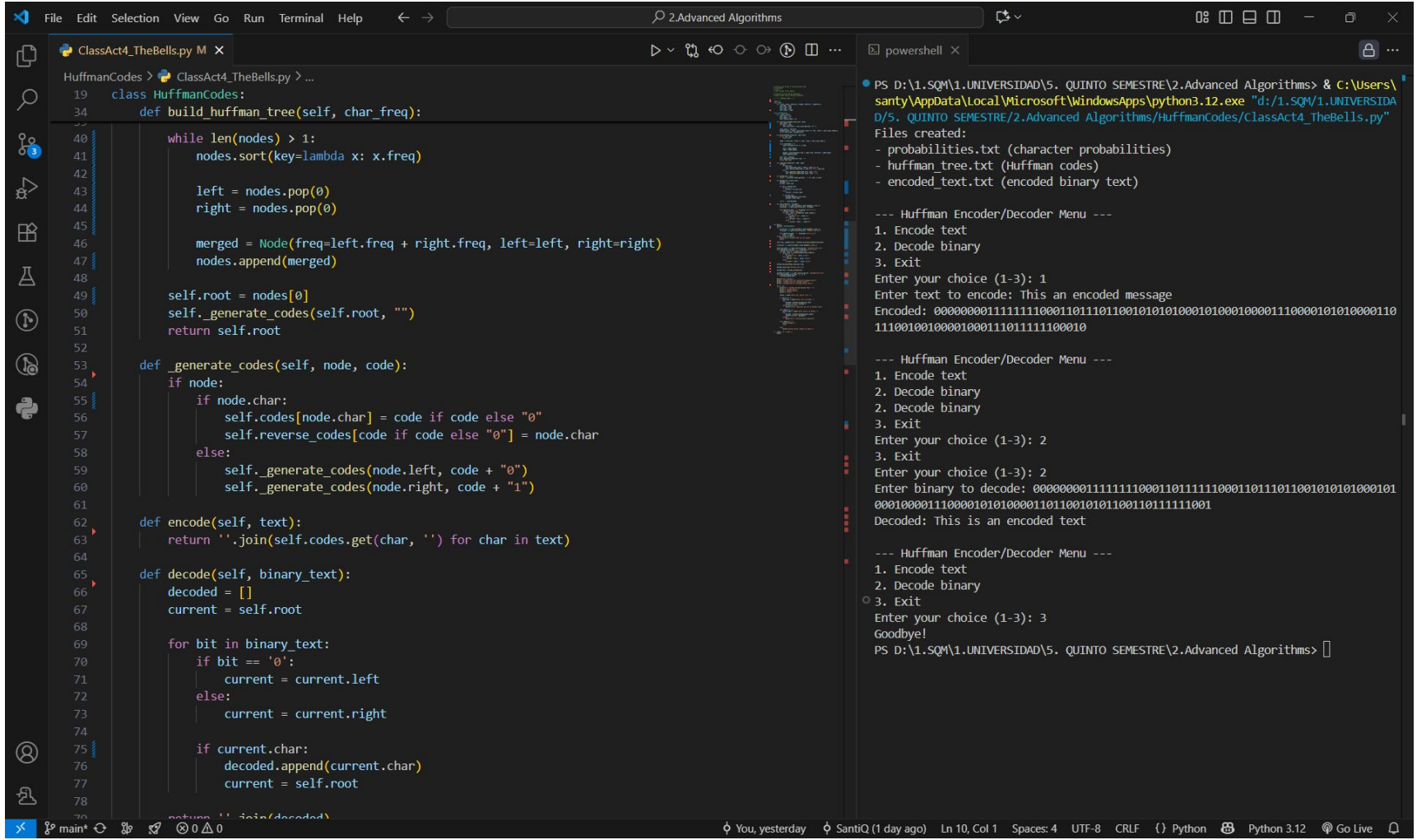
Tecnológico de Monterrey - Campus Monterrey
School of Engineering and Sciences
Engineering in Computational Technologies
Analysis and Design of Advanced Algorithms

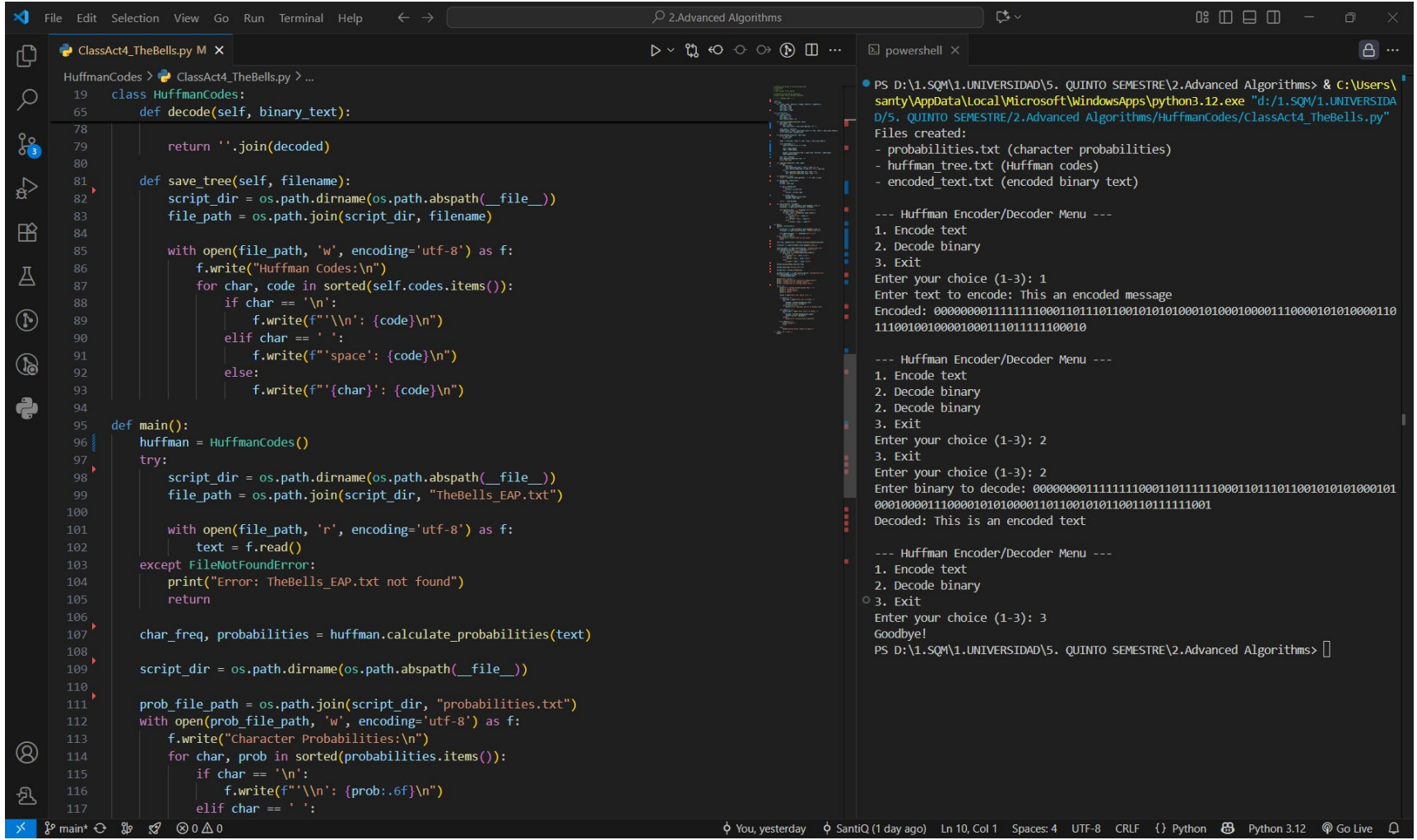
Class Activity 4:
Huffman Codes

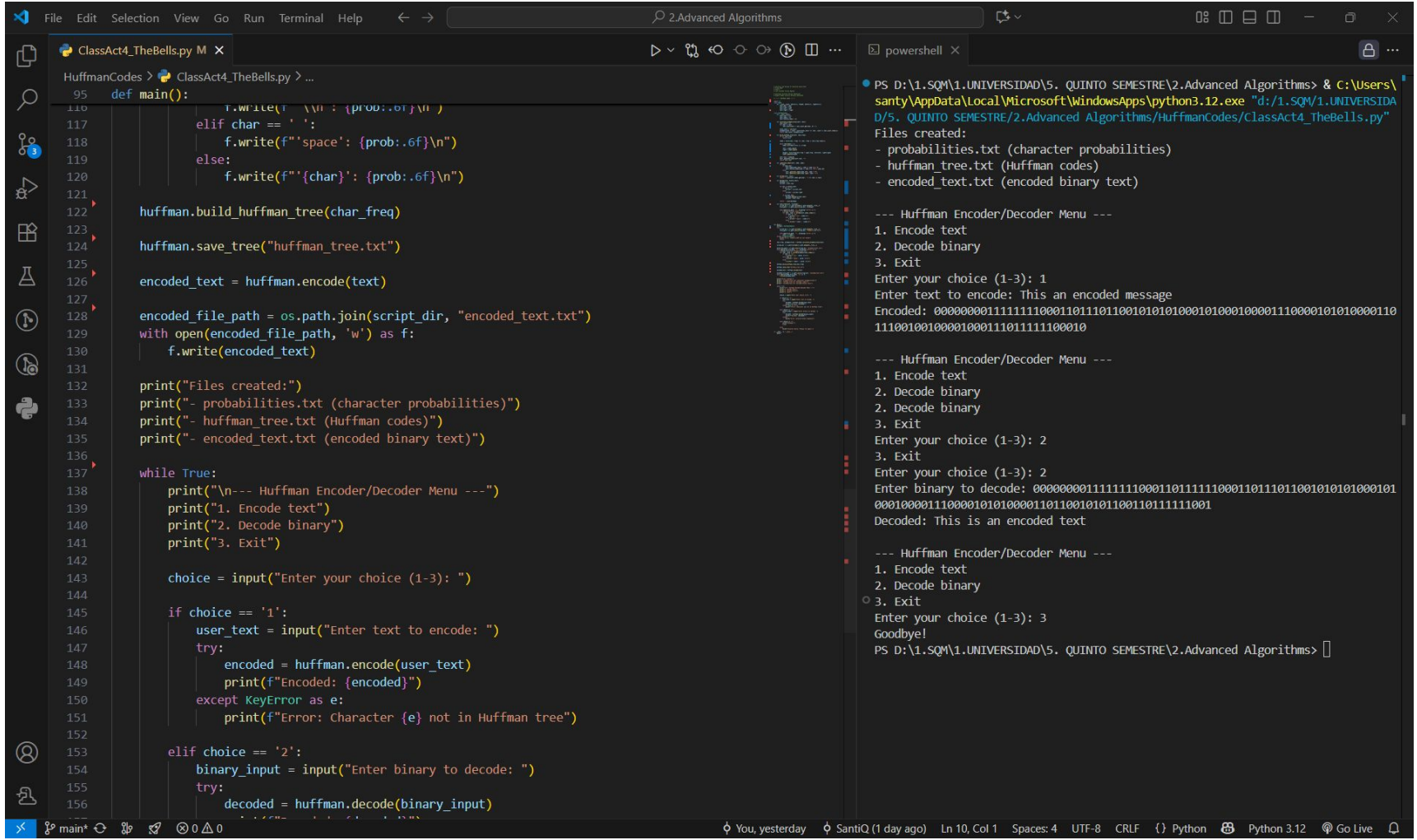
Group: 607
Team #3
Luis Salomón Flores Ugalde

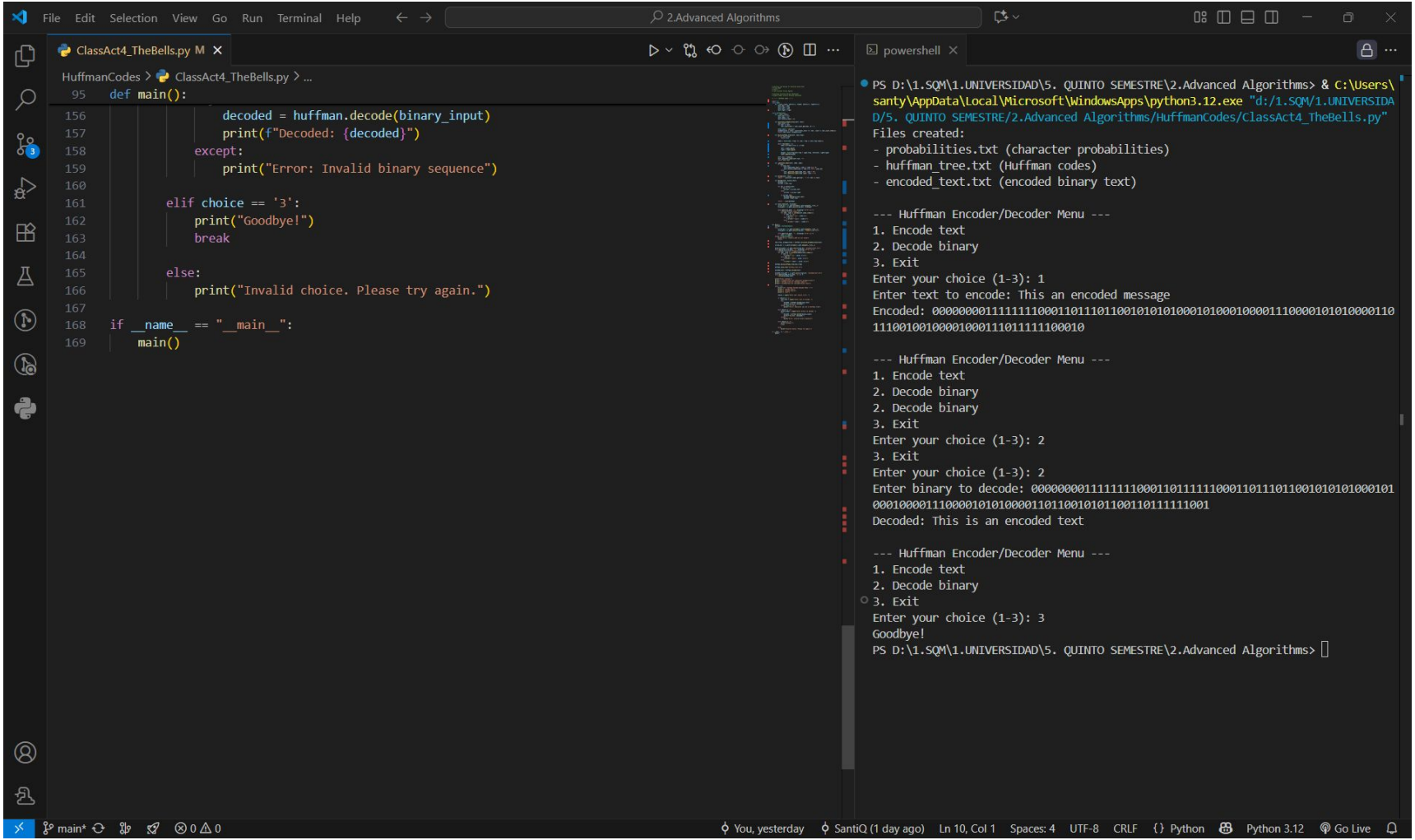
Santiago Quintana Moreno A01571222
Miguel Ángel Álvarez Hermida a01722925

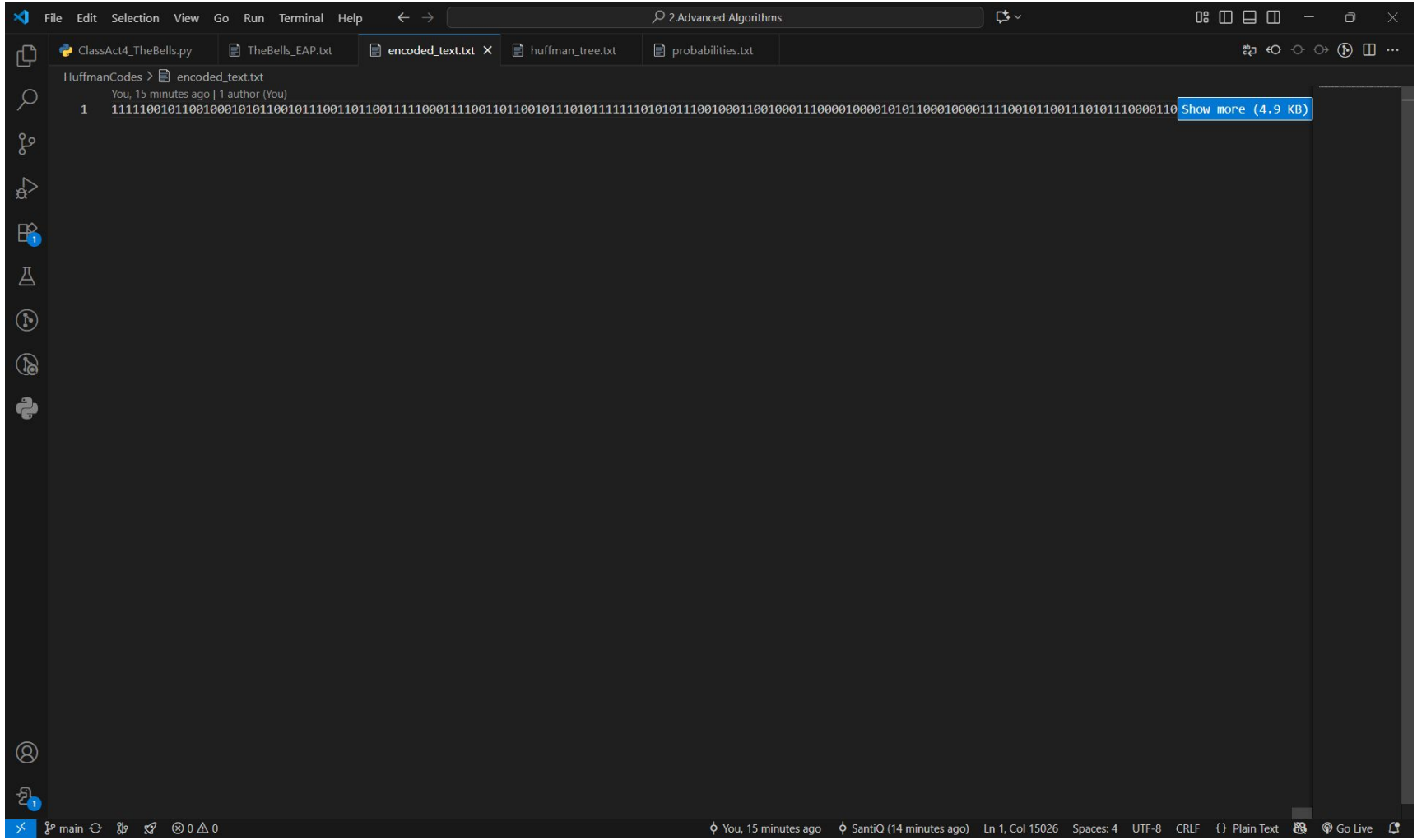

File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
ClassAct4_TheBells.py M
HuffmanCodes > ClassAct4_TheBells.py > ...
You, 2 minutes ago | 1 author (You)
# Analysis and Design of Advanced Algorithms
# Group #607
# Team 3
# Luis Salomón Flores Ugalde
# Santiago Quintana Moreno A01571222
# Miguel Ángel Álvarez Hermida A01722925
# ------ HUFFMAN CODES ------
import os
You, yesterday | 1 author (You)
class Node:
def __init__(self, char=None, freq=0, left=None, right=None):
self.char = char
self.freq = freq
self.left = left
self.right = right
You, 3 minutes ago | 1 author (You)
class HuffmanCodes:
def __init__(self):
self.root = None
self.codes = {}
self.reverse_codes = {}
def calculate_probabilities(self, text):
char_count = {}
for char in text:
char_count[char] = char_count.get(char, 0) + 1
total_chars = len(text)
probabilities = {char: count/total_chars for char, count in char_count.items()}
return char_count, probabilities
def build_huffman_tree(self, char_freq):
if not char_freq:
return None
nodes = [Node(char, freq) for char, freq in char_freq.items()]
powershell
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & C:\Users\santy\AppData\Local\Microsoft\WindowsApps\python3.12.exe "d:/1.SQM/1.UNIVERSIDAD/5. QUINTO SEMESTRE/2.Advanced Algorithms/HuffmanCodes/ClassAct4_TheBells.py"
Files created:
- probabilities.txt (character probabilities)
- huffman_tree.txt (Huffman codes)
- encoded_text.txt (encoded binary text)
--- Huffman Encoder/Decoder Menu ---
1. Encode text
2. Decode binary
3. Exit
Enter your choice (1-3): 1
Enter text to encode: This an encoded message
Encoded: 0000000011111111000110111011001010101000101000100001110000101010000110111001001000010001110111111100010
--- Huffman Encoder/Decoder Menu ---
1. Encode text
2. Decode binary
2. Decode binary
3. Exit
Enter your choice (1-3): 2
3. Exit
Enter your choice (1-3): 2
Enter binary to decode: 0000000011111111000110111011100011011101100101010100010100010000111000010101000011011001010110011011111100 1
Decoded: This is an encoded text
--- Huffman Encoder/Decoder Menu ---
1. Encode text
2. Decode binary
3. Exit
Enter your choice (1-3): 3
Goodbye!
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>
main*
0 0
Go Live


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
ClassAct4_TheBells.py M
HuffmanCodes > ClassAct4_TheBells.py > ...
class HuffmanCodes:
def build_huffman_tree(self, char_freq):
while len(nodes) > 1:
nodes.sort(key=lambda x: x.freq)
left = nodes.pop(0)
right = nodes.pop(0)
merged = Node(freq=left.freq + right.freq, left=left, right=right)
nodes.append(merged)
self.root = nodes[0]
self._generate_codes(self.root, "")
return self.root
def _generate_codes(self, node, code):
if node:
if node.char:
self.codes[node.char] = code if code else "0"
self.reverse_codes[code if code else "0"] = node.char
else:
self._generate_codes(node.left, code + "0")
self._generate_codes(node.right, code + "1")
def encode(self, text):
return ''.join(self.codes.get(char, '') for char in text)
def decode(self, binary_text):
decoded = []
current = self.root
for bit in binary_text:
if bit == '0':
current = current.left
else:
current = current.right
if current.char:
decoded.append(current.char)
current = self.root
powershell
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & C:\Users\santy\AppData\Local\Microsoft\WindowsApps\python3.12.exe "d:/1.SQM/1.UNIVERSIDAD/5. QUINTO SEMESTRE/2.Advanced Algorithms/HuffmanCodes/ClassAct4_TheBells.py"
Files created:
- probabilities.txt (character probabilities)
- huffman_tree.txt (Huffman codes)
- encoded_text.txt (encoded binary text)
--- Huffman Encoder/Decoder Menu ---
1. Encode text
2. Decode binary
3. Exit
Enter your choice (1-3): 1
Enter text to encode: This an encoded message
Encoded: 0000000011111111000110111011001010101000101000100001110000101010000110111100100100001000111011111100010
--- Huffman Encoder/Decoder Menu ---
1. Encode text
2. Decode binary
2. Decode binary
3. Exit
Enter your choice (1-3): 2
3. Exit
Enter your choice (1-3): 2
Enter binary to decode: 0000000011111111000110111111000110111011001010101000101000100011100001010100001101100101011001101111111001
Decoded: This is an encoded text
--- Huffman Encoder/Decoder Menu ---
1. Encode text
2. Decode binary
3. Exit
Enter your choice (1-3): 3
Goodbye!
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>
main*
You, yesterday
SantiQ (1 day ago)
Ln 10, Col 1
Spaces: 4
UTF-8
CRLF
Python
Python 3.12
Go Live


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
ClassAct4_TheBells.py M
HuffmanCodes > ClassAct4_TheBells.py > ...
class HuffmanCodes:
def decode(self, binary_text):
return ''.join(decoded)
def save_tree(self, filename):
script_dir = os.path.dirname(os.path.abspath(__file__))
file_path = os.path.join(script_dir, filename)
with open(file_path, 'w', encoding='utf-8') as f:
f.write("Huffman Codes:\n")
for char, code in sorted(self.codes.items()):
if char == '\n':
f.write(f"'\\n': {code}\n")
elif char == ' ':
f.write(f"'space': {code}\n")
else:
f.write(f"'{char}': {code}\n")
def main():
huffman = HuffmanCodes()
try:
script_dir = os.path.dirname(os.path.abspath(__file__))
file_path = os.path.join(script_dir, "TheBells_EAP.txt")
with open(file_path, 'r', encoding='utf-8') as f:
text = f.read()
except FileNotFoundError:
print("Error: TheBells_EAP.txt not found")
return
char_freq, probabilities = huffman.calculate_probabilities(text)
script_dir = os.path.dirname(os.path.abspath(__file__))
prob_file_path = os.path.join(script_dir, "probabilities.txt")
with open(prob_file_path, 'w', encoding='utf-8') as f:
f.write("Character Probabilities:\n")
for char, prob in sorted(probabilities.items()):
if char == '\n':
f.write(f"'\\n': {prob:.6f}\n")
elif char == ' ':
powershell
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & C:\Users\santy\AppData\Local\Microsoft\WindowsApps\python3.12.exe "d:/1.SQM/1.UNIVERSIDAD/5. QUINTO SEMESTRE/2.Advanced Algorithms/HuffmanCodes/ClassAct4_TheBells.py"
Files created:
- probabilities.txt (character probabilities)
- huffman_tree.txt (Huffman codes)
- encoded_text.txt (encoded binary text)
--- Huffman Encoder/Decoder Menu ---
1. Encode text
2. Decode binary
3. Exit
Enter your choice (1-3): 1
Enter text to encode: This an encoded message
Encoded: 0000000011111111000110111011001010101000101000100001110000101010000110111001001000010001110111111100010
--- Huffman Encoder/Decoder Menu ---
1. Encode text
2. Decode binary
2. Decode binary
3. Exit
Enter your choice (1-3): 2
3. Exit
Enter your choice (1-3): 2
Enter binary to decode: 0000000011111111000110111111000110111011000101010100010100010000111000010101000011011001010110011011111100 1
Decoded: This is an encoded text
--- Huffman Encoder/Decoder Menu ---
1. Encode text
2. Decode binary
3. Exit
Enter your choice (1-3): 3
Goodbye!
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>
main*
0 0
You, yesterday
SantiQ (1 day ago)
Ln 10, Col 1
Spaces: 4
UTF-8
CRLF
Python
Python 3.12
Go Live


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
ClassAct4_TheBells.py M
HuffmanCodes > ClassAct4_TheBells.py > ...
def main():
elif char == ' ':
f.write(f"'space': {prob:.6f}\n")
else:
f.write(f"'{char}': {prob:.6f}\n")
huffman.build_huffman_tree(char_freq)
huffman.save_tree("huffman_tree.txt")
encoded_text = huffman.encode(text)
encoded_file_path = os.path.join(script_dir, "encoded_text.txt")
with open(encoded_file_path, 'w') as f:
f.write(encoded_text)
print("Files created:")
print("- probabilities.txt (character probabilities)")
print("- huffman_tree.txt (Huffman codes)")
print("- encoded_text.txt (encoded binary text)")
while True:
print("\n--- Huffman Encoder/Decoder Menu ---")
print("1. Encode text")
print("2. Decode binary")
print("3. Exit")
choice = input("Enter your choice (1-3): ")
if choice == '1':
user_text = input("Enter text to encode: ")
try:
encoded = huffman.encode(user_text)
print(f"Encoded: {encoded}")
except KeyError as e:
print(f"Error: Character {e} not in Huffman tree")
elif choice == '2':
binary_input = input("Enter binary to decode: ")
try:
decoded = huffman.decode(binary_input)
powershell
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & C:\Users\santy\AppData\Local\Microsoft\WindowsApps\python3.12.exe "d:/1.SQM/1.UNIVERSIDAD/5. QUINTO SEMESTRE/2.Advanced Algorithms/HuffmanCodes/ClassAct4_TheBells.py"
Files created:
- probabilities.txt (character probabilities)
- huffman_tree.txt (Huffman codes)
- encoded_text.txt (encoded binary text)
--- Huffman Encoder/Decoder Menu ---
1. Encode text
2. Decode binary
3. Exit
Enter your choice (1-3): 1
Enter text to encode: This an encoded message
Encoded: 0000000011111111000110111011001010101000101000100001110000101010000110111100100100001000111011111100010
--- Huffman Encoder/Decoder Menu ---
1. Encode text
2. Decode binary
2. Decode binary
3. Exit
Enter your choice (1-3): 2
3. Exit
Enter your choice (1-3): 2
Enter binary to decode: 0000000011111111000110111111000110111011000101010100010100010000111000010101000011011001010110011011111100 1
Decoded: This is an encoded text
--- Huffman Encoder/Decoder Menu ---
1. Encode text
2. Decode binary
3. Exit
Enter your choice (1-3): 3
Goodbye!
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>
main*
0 0
You, yesterday
SantiQ (1 day ago)
Ln 10, Col 1
Spaces: 4
UTF-8
CRLF
Python
Python 3.12
Go Live


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
ClassAct4_TheBells.py M
HuffmanCodes > ClassAct4_TheBells.py > ...
95 def main():
156 decoded = huffman.decode(binary_input)
157 print(f"Decoded: {decoded}")
158 except:
159 print("Error: Invalid binary sequence")
160
161 elif choice == '3':
162 print("Goodbye!")
163 break
164
165 else:
166 print("Invalid choice. Please try again.")
167
168 if __name__ == "__main__":
169 main()
powershell
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & C:\Users\santy\AppData\Local\Microsoft\WindowsApps\python3.12.exe "d:/1.SQM/1.UNIVERSIDAD/5. QUINTO SEMESTRE/2.Advanced Algorithms/HuffmanCodes/ClassAct4_TheBells.py"
Files created:
- probabilities.txt (character probabilities)
- huffman_tree.txt (Huffman codes)
- encoded_text.txt (encoded binary text)
--- Huffman Encoder/Decoder Menu ---
1. Encode text
2. Decode binary
3. Exit
Enter your choice (1-3): 1
Enter text to encode: This an encoded message
Encoded: 00000000111111110001101110110010101010001010001000011100001010100001101111001001000010001110111111100010
--- Huffman Encoder/Decoder Menu ---
1. Encode text
2. Decode binary
2. Decode binary
3. Exit
Enter your choice (1-3): 2
3. Exit
Enter your choice (1-3): 2
Enter binary to decode: 00000000111111110001101111110001101110110010101010001010001000011100001010100001101100101011001101111111001
Decoded: This is an encoded text
--- Huffman Encoder/Decoder Menu ---
1. Encode text
2. Decode binary
3. Exit
Enter your choice (1-3): 3
Goodbye!
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>
main*
0 0
You, yesterday
SantiQ (1 day ago)
Ln 10, Col 1
Spaces: 4
UTF-8
CRLF
Python
Python 3.12
Go Live


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
ClassAct4_TheBells.py
TheBells_EAP.txt
encoded_text.txt
huffman_tree.txt
probabilities.txt
HuffmanCodes > encoded_text.txt
You, 15 minutes ago | 1 author (You)
1
Show more (4.9 KB)
main
0 0
You, 15 minutes ago
SantiQ (14 minutes ago)
Ln 1, Col 15026
Spaces: 4
UTF-8
CRLF
Plain Text
Go Live

File Edit Selection View Go Run Terminal Help

2.Advanced Algorithms

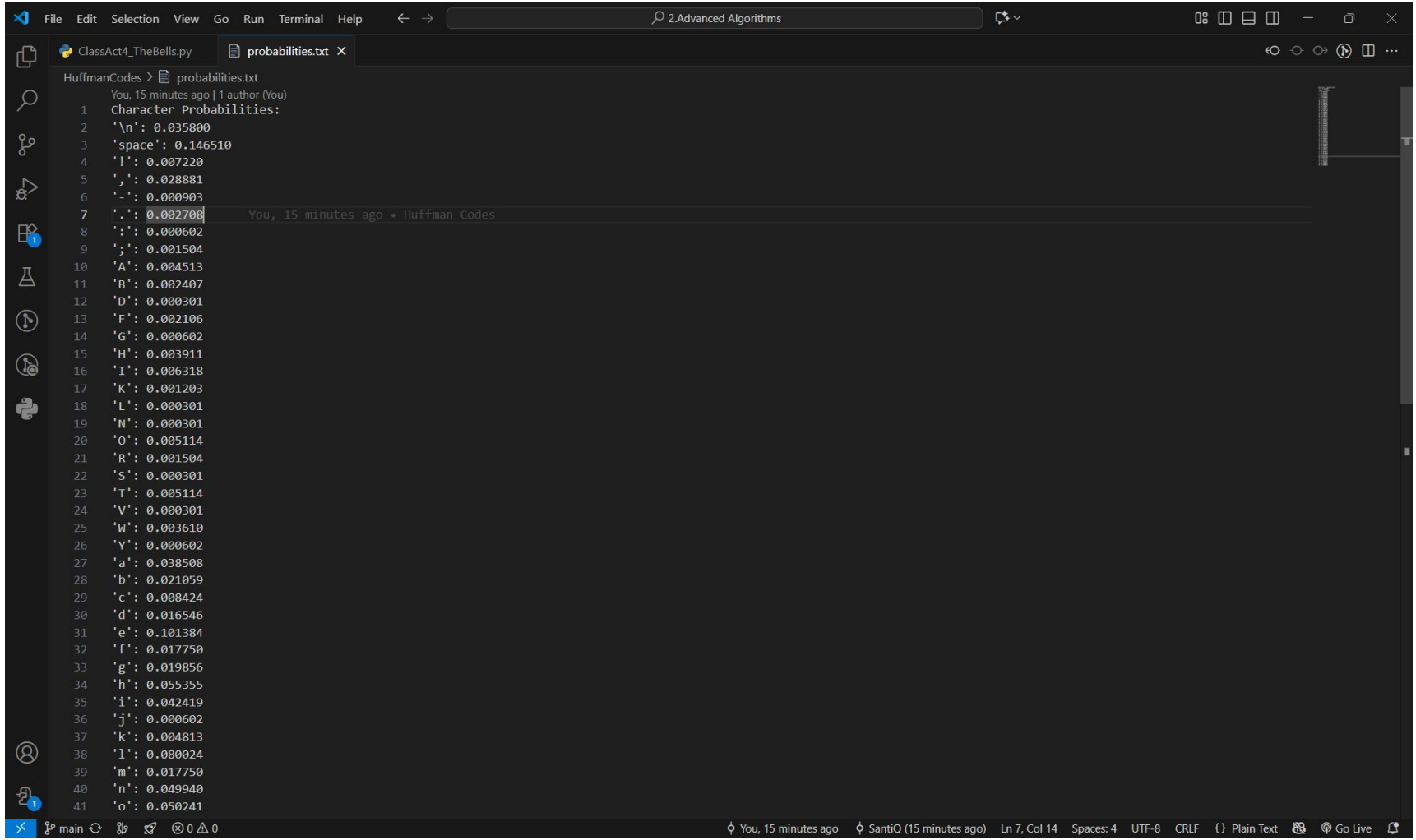
ClassAct4_TheBells.py | huffman_tree.txt | probabilities.txt

HuffmanCodes > huffman_tree.txt

You, 15 minutes ago | 1 author (You)

```
Huffman Codes:      You, 15 minutes ago • Huffman Codes
'\n': 11000
'space': 101
'!': 1000000
',': 01101
'-': 0000011111
'.': 00000100
':': 10001010100
';': 000001110
'A': 10001011
'B': 110011011
'D': 100010101101
'F': 100010100
'G': 11001101010
'H': 10000011
'I': 0110001
'K': 000001010
'L': 100010101111
'N': 100010101110
'O': 0000000
'R': 000001011
'S': 110011010011
'T': 0000001
'V': 100010101100
'W': 10000010
'Y': 10001010101
'a': 11011
'b': 111101
'c': 1000100
'd': 100001
'e': 010
'f': 100011
'g': 111100
'h': 0111
'i': 11111
'j': 11001101011
'k': 11001100
'l': 1110
'm': 110010
'n': 0010
'o': 0011
```

main ⊗0 ⚠0 | You, 15 minutes ago | Ln 1, Col 1 | Spaces: 4 | UTF-8 | CRLF | { } Plain Text | Go Live


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
ClassAct4_TheBells.py
probabilities.txt
HuffmanCodes > probabilities.txt
You, 15 minutes ago | 1 author (You)
Character Probabilities:
'\n': 0.035800
'space': 0.146510
'!': 0.007220
',': 0.028881
'-': 0.000903
'.': 0.002708
You, 15 minutes ago • Huffman Codes
':': 0.000602
';': 0.001504
'A': 0.004513
'B': 0.002407
'D': 0.000301
'F': 0.002106
'G': 0.000602
'H': 0.003911
'I': 0.006318
'K': 0.001203
'L': 0.000301
'N': 0.000301
'O': 0.005114
'R': 0.001504
'S': 0.000301
'T': 0.005114
'V': 0.000301
'W': 0.003610
'Y': 0.000602
'a': 0.038508
'b': 0.021059
'c': 0.008424
'd': 0.016546
'e': 0.101384
'f': 0.017750
'g': 0.019856
'h': 0.055355
'i': 0.042419
'j': 0.000602
'k': 0.004813
'l': 0.080024
'm': 0.017750
'n': 0.049940
'o': 0.050241
main
0 0
You, 15 minutes ago
SantiQ (15 minutes ago)
Ln 7, Col 14
Spaces: 4
UTF-8
CRLF
Plain Text
Go Live

https://colab.research.google.com/drive/1GAjJSPmSfVweoDj5X2JqajzWx-SEF1tX?usp=sharing

REFERENCES

GeeksforGeeks. (2025, July 23). *Huffman Coding | Greedy algo3*. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/huffman-coding-greedy-algo-3/

GeeksforGeeks. (2025, July 23). *Huffman Coding in Python*. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/huffman-coding-in-python/

W3Schools.com. (n.d.). https://www.w3schools.com/dsa/dsa_ref_huffman_coding.php