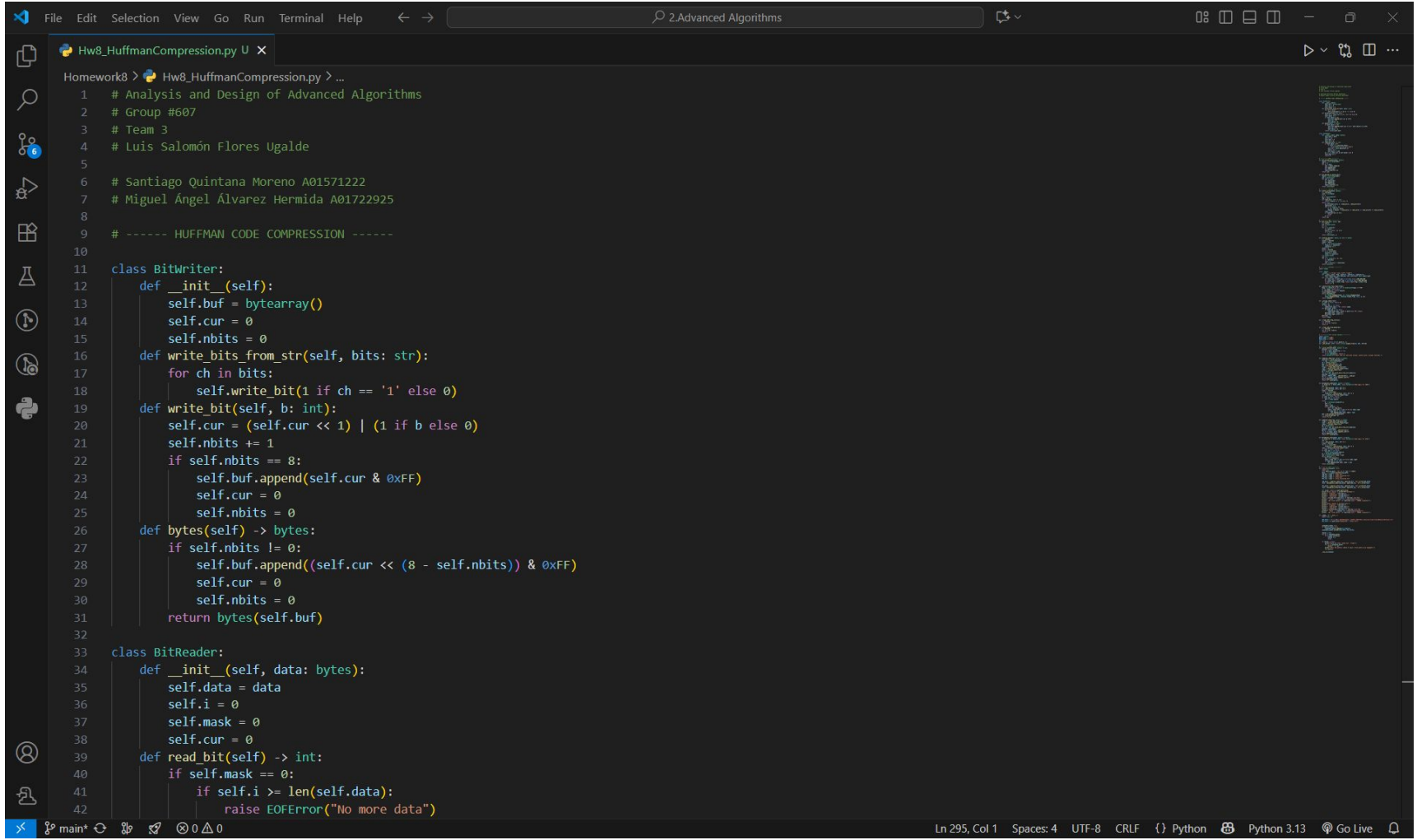
Tecnológico de Monterrey

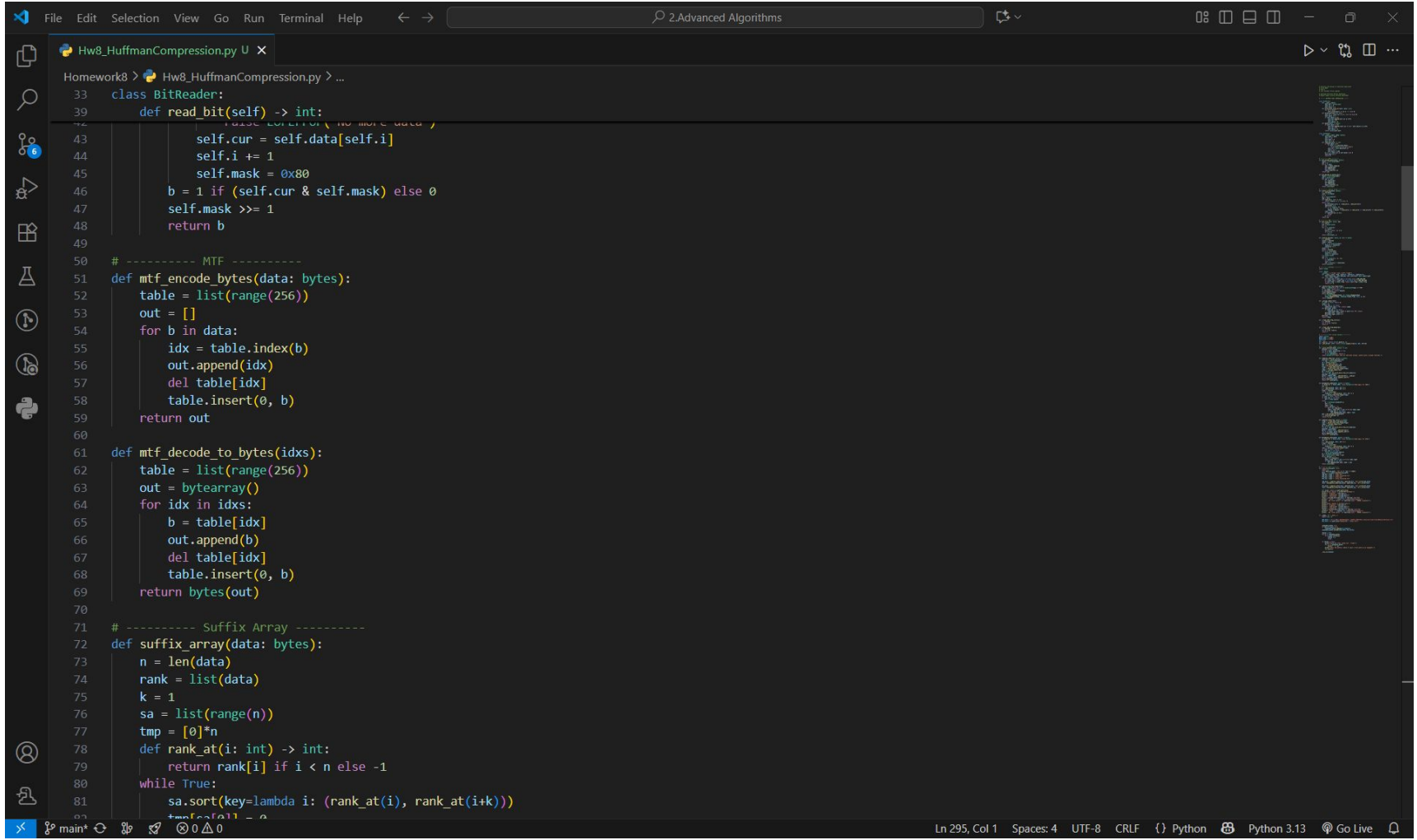
Tecnológico de Monterrey - Campus Monterrey
School of Engineering and Sciences
Engineering in Computational Technologies
Analysis and Design of Advanced Algorithms

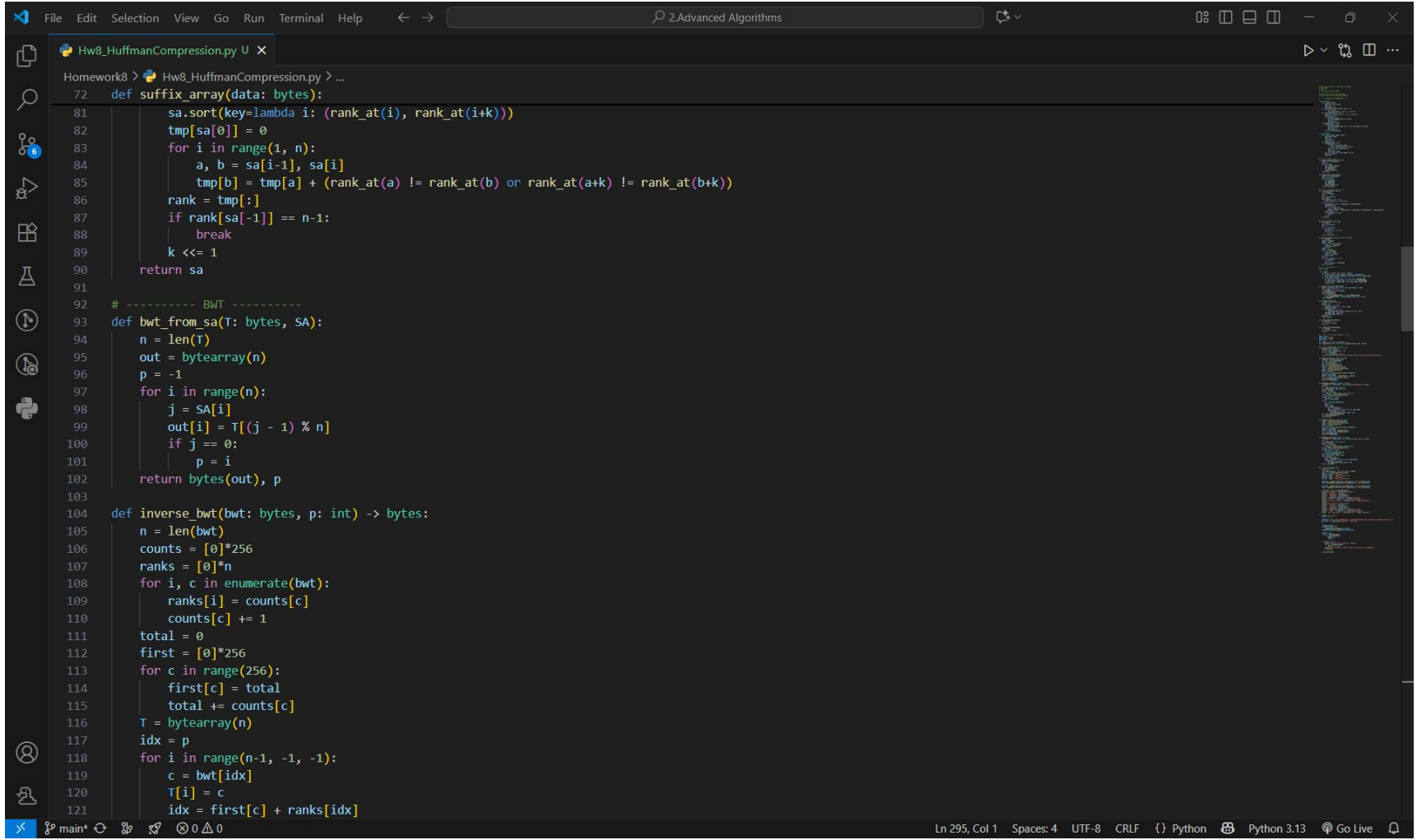
Homework 8:
Huffman Compression

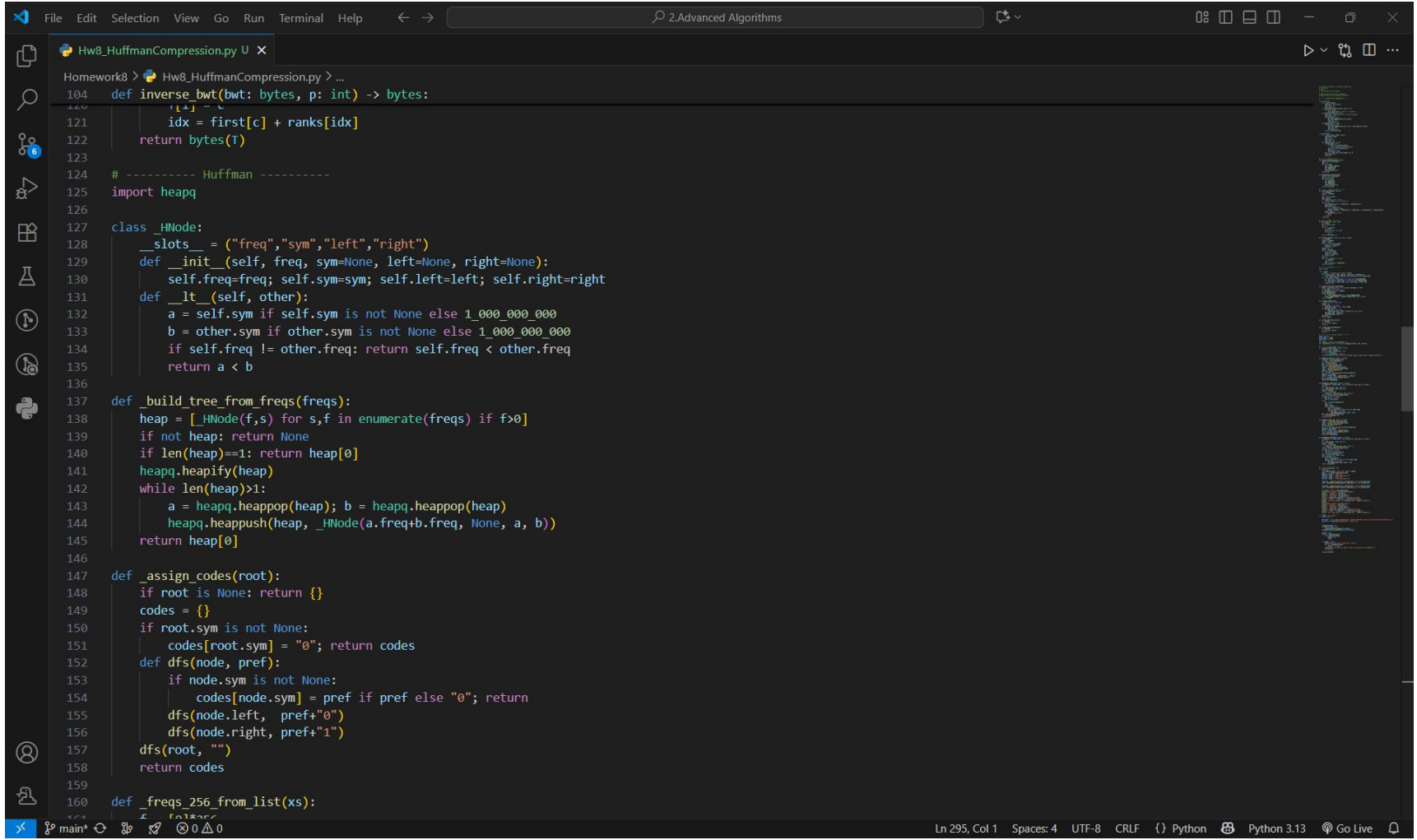
Group: 607
Team #3
Luis Salomón Flores Ugalde

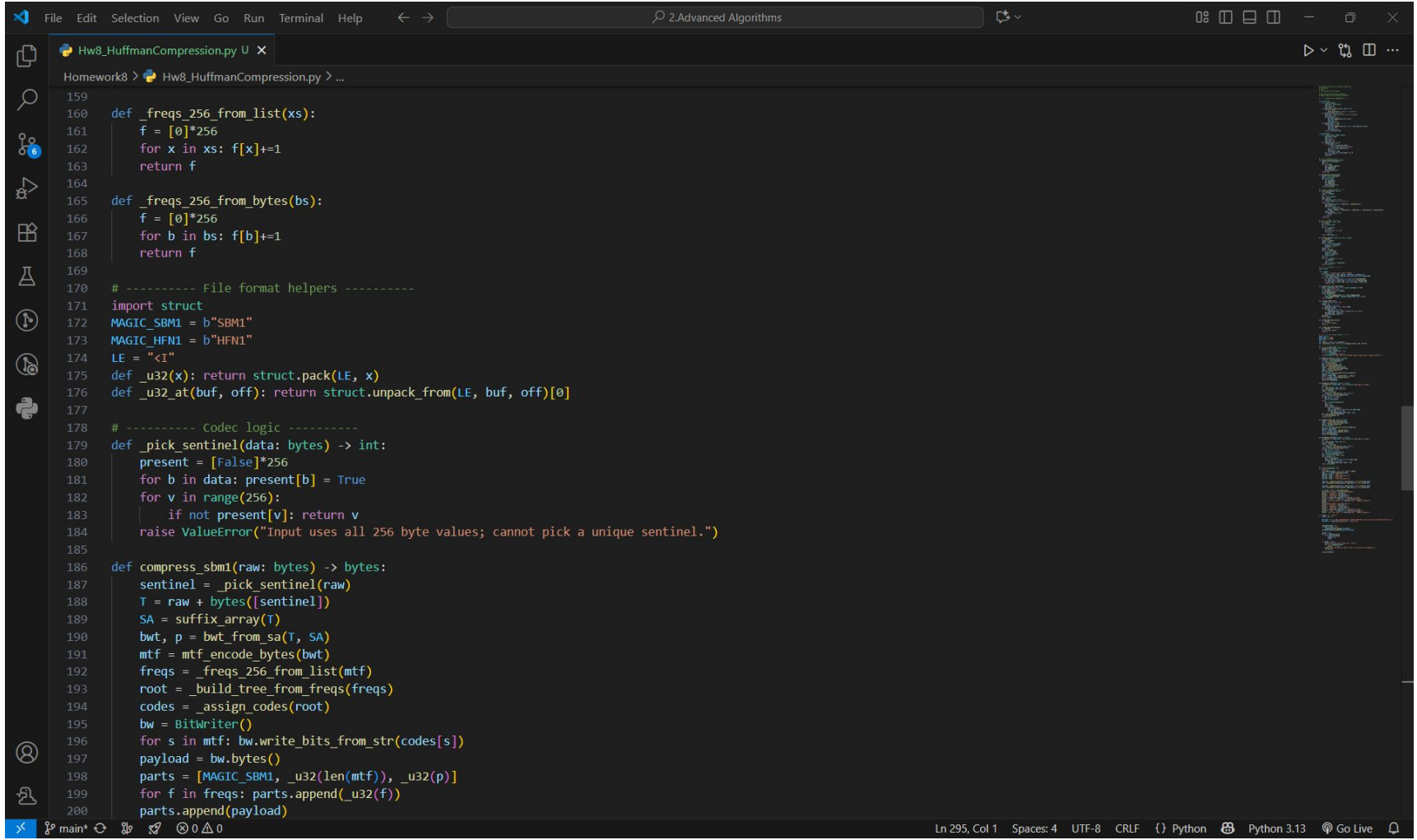
Santiago Quintana Moreno A01571222
Miguel Ángel Álvarez Hermida a01722925

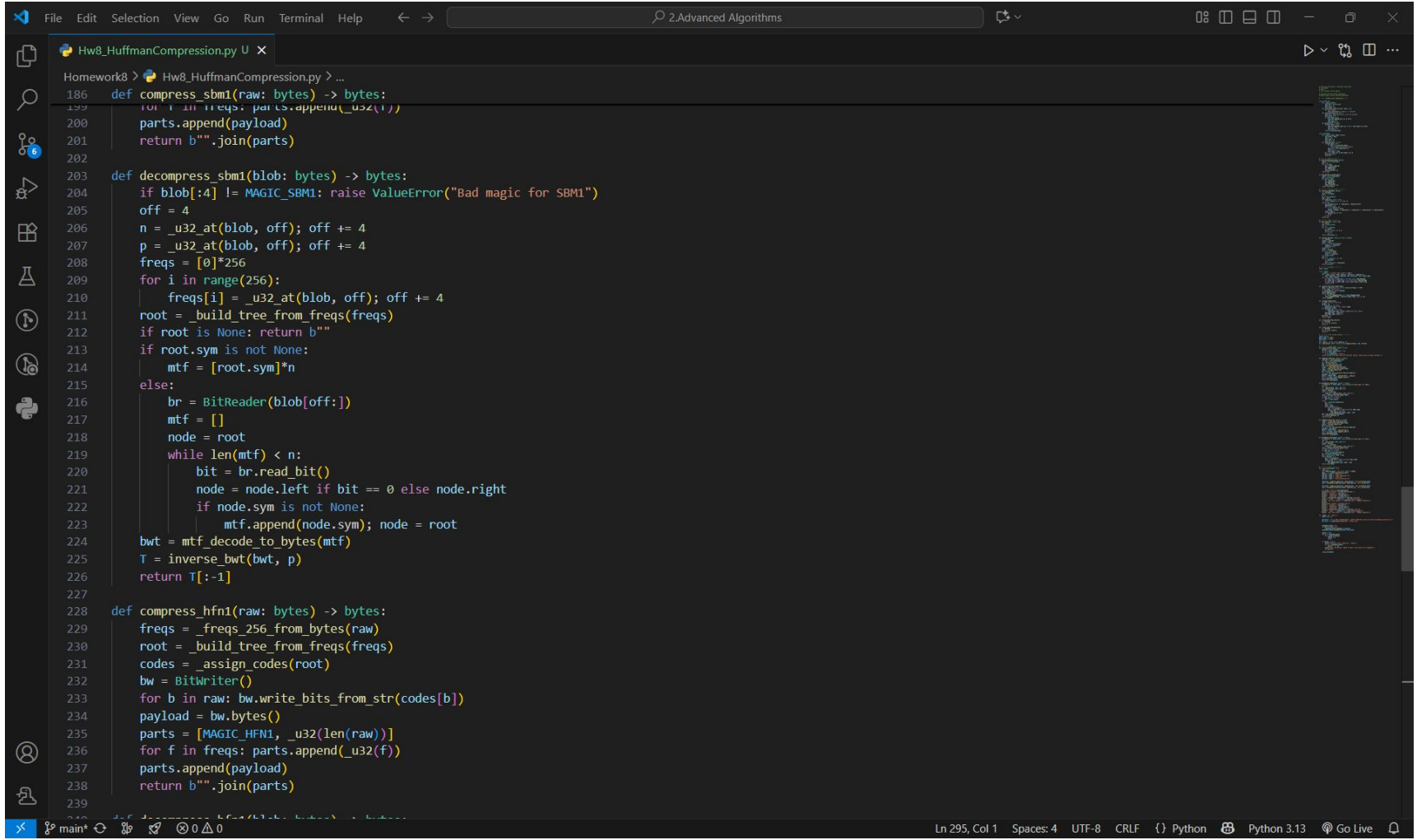

File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
Hw8_HuffmanCompression.py U
Homework8 > Hw8_HuffmanCompression.py > ...
# Analysis and Design of Advanced Algorithms
# Group #607
# Team 3
# Luis Salomón Flores Ugalde
# Santiago Quintana Moreno A01571222
# Miguel Ángel Álvarez Hermida A01722925
# ------ HUFFMAN CODE COMPRESSION ------
class BitWriter:
def __init__(self):
self.buf = bytearray()
self.cur = 0
self.nbits = 0
def write_bits_from_str(self, bits: str):
for ch in bits:
self.write_bit(1 if ch == '1' else 0)
def write_bit(self, b: int):
self.cur = (self.cur << 1) | (1 if b else 0)
self.nbits += 1
if self.nbits == 8:
self.buf.append(self.cur & 0xFF)
self.cur = 0
self.nbits = 0
def bytes(self) -> bytes:
if self.nbits != 0:
self.buf.append((self.cur << (8 - self.nbits)) & 0xFF)
self.cur = 0
self.nbits = 0
return bytes(self.buf)
class BitReader:
def __init__(self, data: bytes):
self.data = data
self.i = 0
self.mask = 0
self.cur = 0
def read_bit(self) -> int:
if self.mask == 0:
if self.i >= len(self.data):
raise EOFError("No more data")
main*
0 0
Ln 295, Col 1 Spaces: 4 UTF-8 CRLF Python Python 3.13 Go Live

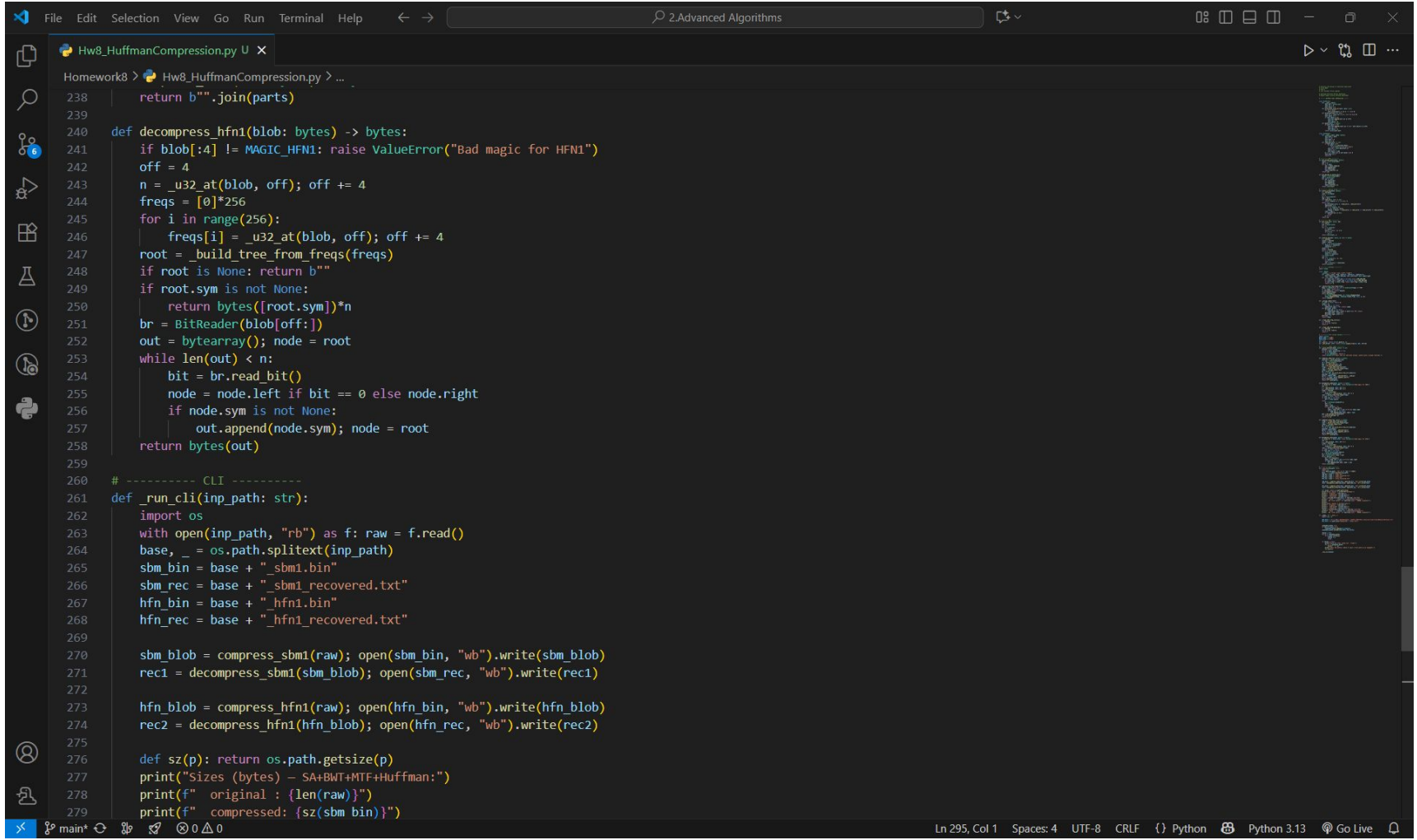

File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
Hw8_HuffmanCompression.py U
Homework8 > Hw8_HuffmanCompression.py > ...
class BitReader:
def read_bit(self) -> int:
self.cur = self.data[self.i]
self.i += 1
self.mask = 0x80
b = 1 if (self.cur & self.mask) else 0
self.mask >>= 1
return b
# ---------- MTF ----------
def mtf_encode_bytes(data: bytes):
table = list(range(256))
out = []
for b in data:
idx = table.index(b)
out.append(idx)
del table[idx]
table.insert(0, b)
return out
def mtf_decode_to_bytes(idxs):
table = list(range(256))
out = bytearray()
for idx in idxs:
b = table[idx]
out.append(b)
del table[idx]
table.insert(0, b)
return bytes(out)
# ---------- Suffix Array ----------
def suffix_array(data: bytes):
n = len(data)
rank = list(data)
k = 1
sa = list(range(n))
tmp = [0]*n
def rank_at(i: int) -> int:
return rank[i] if i < n else -1
while True:
sa.sort(key=lambda i: (rank_at(i), rank_at(i+k)))
main*
0 0
Ln 295, Col 1
Spaces: 4
UTF-8
CRLF
Python
Python 3.13
Go Live

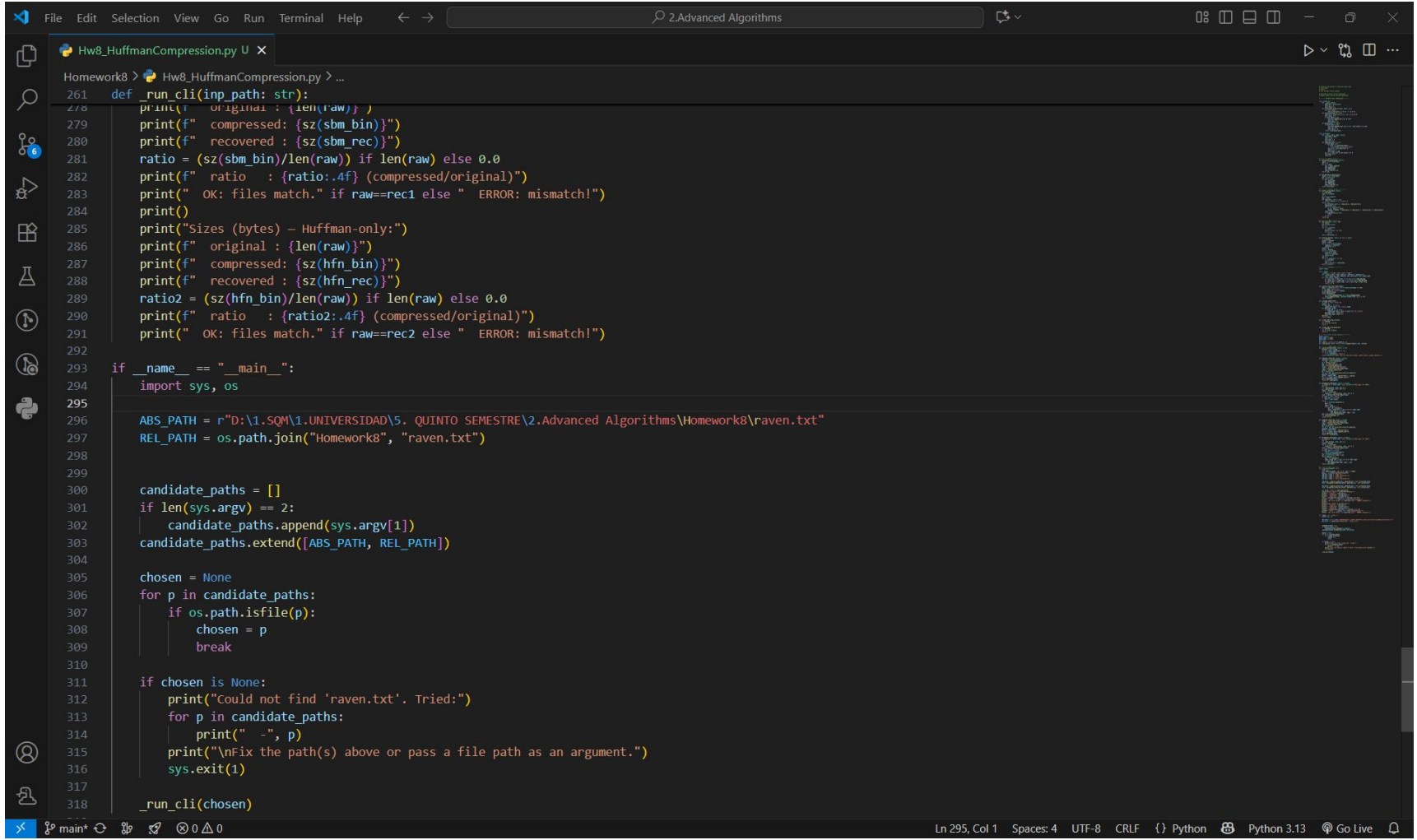

File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
Hw8_HuffmanCompression.py U
Homework8 > Hw8_HuffmanCompression.py > ...
72 def suffix_array(data: bytes):
81 sa.sort(key=lambda i: (rank_at(i), rank_at(i+k)))
82 tmp[sa[0]] = 0
83 for i in range(1, n):
84 a, b = sa[i-1], sa[i]
85 tmp[b] = tmp[a] + (rank_at(a) != rank_at(b) or rank_at(a+k) != rank_at(b+k))
86 rank = tmp[:]
87 if rank[sa[-1]] == n-1:
88 break
89 k <<= 1
90 return sa
91
92 # ---------- BWT ----------
93 def bwt_from_sa(T: bytes, SA):
94 n = len(T)
95 out = bytearray(n)
96 p = -1
97 for i in range(n):
98 j = SA[i]
99 out[i] = T[(j - 1) % n]
100 if j == 0:
101 p = i
102 return bytes(out), p
103
104 def inverse_bwt(bwt: bytes, p: int) -> bytes:
105 n = len(bwt)
106 counts = [0]*256
107 ranks = [0]*n
108 for i, c in enumerate(bwt):
109 ranks[i] = counts[c]
110 counts[c] += 1
111 total = 0
112 first = [0]*256
113 for c in range(256):
114 first[c] = total
115 total += counts[c]
116 T = bytearray(n)
117 idx = p
118 for i in range(n-1, -1, -1):
119 c = bwt[idx]
120 T[i] = c
121 idx = first[c] + ranks[idx]
main*
0 0
Ln 295, Col 1
Spaces: 4
UTF-8
CRLF
Python
Python 3.13
Go Live


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
Hw8_HuffmanCompression.py U
Homework8 > Hw8_HuffmanCompression.py > ...
104 def inverse_bwt(bwt: bytes, p: int) -> bytes:
121 idx = first[c] + ranks[idx]
122 return bytes(T)
124 # ---------- Huffman ----------
125 import heapq
127 class _HNode:
128 __slots__ = ("freq","sym","left","right")
129 def __init__(self, freq, sym=None, left=None, right=None):
130 self.freq=freq; self.sym=sym; self.left=left; self.right=right
131 def __lt__(self, other):
132 a = self.sym if self.sym is not None else 1_000_000_000
133 b = other.sym if other.sym is not None else 1_000_000_000
134 if self.freq != other.freq: return self.freq < other.freq
135 return a < b
137 def _build_tree_from_freqs(freqs):
138 heap = [_HNode(f,s) for s,f in enumerate(freqs) if f>0]
139 if not heap: return None
140 if len(heap)==1: return heap[0]
141 heapq.heapify(heap)
142 while len(heap)>1:
143 a = heapq.heappop(heap); b = heapq.heappop(heap)
144 heapq.heappush(heap, _HNode(a.freq+b.freq, None, a, b))
145 return heap[0]
147 def _assign_codes(root):
148 if root is None: return {}
149 codes = {}
150 if root.sym is not None:
151 codes[root.sym] = "0"; return codes
152 def dfs(node, pref):
153 if node.sym is not None:
154 codes[node.sym] = pref if pref else "0"; return
155 dfs(node.left, pref+"0")
156 dfs(node.right, pref+"1")
157 dfs(root, "")
158 return codes
160 def _freqs_256_from_list(xs):
main*
0 0
Ln 295, Col 1 Spaces: 4 UTF-8 CRLF {} Python Python 3.13 Go Live


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
Hw8_HuffmanCompression.py U
Homework8 > Hw8_HuffmanCompression.py > ...
159
160 def _freqs_256_from_list(xs):
161 f = [0]*256
162 for x in xs: f[x]+=1
163 return f
164
165 def _freqs_256_from_bytes(bs):
166 f = [0]*256
167 for b in bs: f[b]+=1
168 return f
169
170 # ---------- File format helpers ----------
171 import struct
172 MAGIC_SBM1 = b"SBM1"
173 MAGIC_HFN1 = b"HFN1"
174 LE = "<I"
175 def _u32(x): return struct.pack(LE, x)
176 def _u32_at(buf, off): return struct.unpack_from(LE, buf, off)[0]
177
178 # ---------- Codec logic ----------
179 def _pick_sentinel(data: bytes) -> int:
180 present = [False]*256
181 for b in data: present[b] = True
182 for v in range(256):
183 if not present[v]: return v
184 raise ValueError("Input uses all 256 byte values; cannot pick a unique sentinel.")
185
186 def compress_sbm1(raw: bytes) -> bytes:
187 sentinel = _pick_sentinel(raw)
188 T = raw + bytes([sentinel])
189 SA = suffix_array(T)
190 bwt, p = bwt_from_sa(T, SA)
191 mtf = mtf_encode_bytes(bwt)
192 freqs = _freqs_256_from_list(mtf)
193 root = _build_tree_from_freqs(freqs)
194 codes = _assign_codes(root)
195 bw = BitWriter()
196 for s in mtf: bw.write_bits_from_str(codes[s])
197 payload = bw.bytes()
198 parts = [MAGIC_SBM1, _u32(len(mtf)), _u32(p)]
199 for f in freqs: parts.append(_u32(f))
200 parts.append(payload)
main*
0 0
Ln 295, Col 1 Spaces: 4 UTF-8 CRLF {} Python Python 3.13 Go Live


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
Hw8_HuffmanCompression.py U
Homework8 > Hw8_HuffmanCompression.py > ...
186 def compress_sbm1(raw: bytes) -> bytes:
200 parts.append(payload)
201 return b"".join(parts)
203 def decompress_sbm1(blob: bytes) -> bytes:
204 if blob[:4] != MAGIC_SBM1: raise ValueError("Bad magic for SBM1")
205 off = 4
206 n = _u32_at(blob, off); off += 4
207 p = _u32_at(blob, off); off += 4
208 freqs = [0]*256
209 for i in range(256):
210 freqs[i] = _u32_at(blob, off); off += 4
211 root = _build_tree_from_freqs(freqs)
212 if root is None: return b""
213 if root.sym is not None:
214 mtf = [root.sym]*n
215 else:
216 br = BitReader(blob[off:])
217 mtf = []
218 node = root
219 while len(mtf) < n:
220 bit = br.read_bit()
221 node = node.left if bit == 0 else node.right
222 if node.sym is not None:
223 mtf.append(node.sym); node = root
224 bwt = mtf_decode_to_bytes(mtf)
225 T = inverse_bwt(bwt, p)
226 return T[:-1]
228 def compress_hfn1(raw: bytes) -> bytes:
229 freqs = _freqs_256_from_bytes(raw)
230 root = _build_tree_from_freqs(freqs)
231 codes = _assign_codes(root)
232 bw = BitWriter()
233 for b in raw: bw.write_bits_from_str(codes[b])
234 payload = bw.bytes()
235 parts = [MAGIC_HFN1, _u32(len(raw))]
236 for f in freqs: parts.append(_u32(f))
237 parts.append(payload)
238 return b"".join(parts)
main*
Ln 295, Col 1 Spaces: 4 UTF-8 CRLF {} Python Python 3.13 Go Live


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
Hw8_HuffmanCompression.py U
Homework8 > Hw8_HuffmanCompression.py > ...
238 return b"".join(parts)
239
240 def decompress_hfn1(blob: bytes) -> bytes:
241 if blob[:4] != MAGIC_HFN1: raise ValueError("Bad magic for HFN1")
242 off = 4
243 n = _u32_at(blob, off); off += 4
244 freqs = [0]*256
245 for i in range(256):
246 freqs[i] = _u32_at(blob, off); off += 4
247 root = _build_tree_from_freqs(freqs)
248 if root is None: return b""
249 if root.sym is not None:
250 return bytes([root.sym])*n
251 br = BitReader(blob[off:])
252 out = bytearray(); node = root
253 while len(out) < n:
254 bit = br.read_bit()
255 node = node.left if bit == 0 else node.right
256 if node.sym is not None:
257 out.append(node.sym); node = root
258 return bytes(out)
259
260 # ---------- CLI ----------
261 def _run_cli(inp_path: str):
262 import os
263 with open(inp_path, "rb") as f: raw = f.read()
264 base, _ = os.path.splitext(inp_path)
265 sbm_bin = base + "_sbm1.bin"
266 sbm_rec = base + "_sbm1_recovered.txt"
267 hfn_bin = base + "_hfn1.bin"
268 hfn_rec = base + "_hfn1_recovered.txt"
269
270 sbm_blob = compress_sbm1(raw); open(sbm_bin, "wb").write(sbm_blob)
271 rec1 = decompress_sbm1(sbm_blob); open(sbm_rec, "wb").write(rec1)
272
273 hfn_blob = compress_hfn1(raw); open(hfn_bin, "wb").write(hfn_blob)
274 rec2 = decompress_hfn1(hfn_blob); open(hfn_rec, "wb").write(rec2)
275
276 def sz(p): return os.path.getsize(p)
277 print("Sizes (bytes) — SA+BWT+MTF+Huffman:")
278 print(f" original : {len(raw)}")
279 print(f" compressed: {sz(sbm bin)}")
main*
0 0
Ln 295, Col 1 Spaces: 4 UTF-8 CRLF {} Python Python 3.13 Go Live


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
Hw8_HuffmanCompression.py U
Homework8 > Hw8_HuffmanCompression.py > ...
261 def _run_cli(inp_path: str):
279 print(f" compressed: {sz(sbm_bin)}")
280 print(f" recovered : {sz(sbm_rec)}")
281 ratio = (sz(sbm_bin)/len(raw)) if len(raw) else 0.0
282 print(f" ratio : {ratio:.4f} (compressed/original)")
283 print(" OK: files match." if raw==rec1 else " ERROR: mismatch!")
284 print()
285 print("Sizes (bytes) — Huffman-only:")
286 print(f" original : {len(raw)}")
287 print(f" compressed: {sz(hfn_bin)}")
288 print(f" recovered : {sz(hfn_rec)}")
289 ratio2 = (sz(hfn_bin)/len(raw)) if len(raw) else 0.0
290 print(f" ratio : {ratio2:.4f} (compressed/original)")
291 print(" OK: files match." if raw==rec2 else " ERROR: mismatch!")
293 if __name__ == "__main__":
294 import sys, os
296 ABS_PATH = r"D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms\Homework8\raven.txt"
297 REL_PATH = os.path.join("Homework8", "raven.txt")
300 candidate_paths = []
301 if len(sys.argv) == 2:
302 candidate_paths.append(sys.argv[1])
303 candidate_paths.extend([ABS_PATH, REL_PATH])
305 chosen = None
306 for p in candidate_paths:
307 if os.path.isfile(p):
308 chosen = p
309 break
311 if chosen is None:
312 print("Could not find 'raven.txt'. Tried:")
313 for p in candidate_paths:
314 print(" -", p)
315 print("\nFix the path(s) above or pass a file path as an argument.")
316 sys.exit(1)
318 _run_cli(chosen)
main* 0 0
Ln 295, Col 1 Spaces: 4 UTF-8 CRLF {} Python Python 3.13 Go Live

File Edit Selection View Go Run Terminal Help

2.Advanced Algorithms

Hw8_HuffmanCompression.py U | raven.txt U | raven_sbm1_recovered.txt U | raven_hfn1_recovered.txt U

Homework8 > raven.txt

```
THE RAVEN by Edgar Allan Poe

Once upon a midnight dreary, while I pondered, weak and weary,
Over many a quaint and curious volume of forgotten lore,
While I nodded, nearly napping, suddenly there came a tapping,
As of some one gently rapping, rapping at my chamber door.
"'Tis some visitor," I muttered, "tapping at my chamber door-
                Only this, and nothing more."

Ah, distinctly I remember it was in the bleak December,
And each separate dying ember wrought its ghost upon the floor.
Eagerly I wished the morrow;- vainly I had sought to borrow
From my books surcease of sorrow- sorrow for the lost Lenore-
For the rare and radiant maiden whom the angels name Lenore-
                Nameless here for evermore.

And the silken, sad, uncertain rustling of each purple curtain
Thrilled me- filled me with fantastic terrors never felt before;
So that now, to still the beating of my heart, I stood repeating,
"'Tis some visitor entreating entrance at my chamber door-
Some late visitor entreating entrance at my chamber door;-
                This it is, and nothing more."

Presently my soul grew stronger; hesitating then no longer,
"Sir," said I, "or Madam, truly your forgiveness I implore;
But the fact is I was napping, and so gently you came rapping,
And so faintly you came tapping, tapping at my chamber door,
That I scarce was sure I heard you"- here I opened wide the door;-
                Darkness there, and nothing more.

Deep into that darkness peering, long I stood there wondering, fearing,
Doubting, dreaming dreams no mortal ever dared to dream before;
But the silence was unbroken, and the stillness gave no token,
And the only word there spoken was the whispered word, "Lenore?"
This I whispered, and an echo murmured back the word, "Lenore!"-
                Merely this, and nothing more.

Back into the chamber turning, all my soul within me burning,
Soon again I heard a tapping somewhat louder than before.
"Surely," said I, "surely that is something at my window lattice:
Let me see, then, what thereat is, and this mystery explore-
Let my heart be still a moment and this mystery explore;-
```

main* ⊗ 0 ⚠ 0 | Ln 22, Col 48 | Spaces: 4 | UTF-8 | LF | {} Plain Text | Go Live

File Edit Selection View Go Run Terminal Help

2.Advanced Algorithms

Hw8_HuffmanCompression.py U | raven.txt U | raven_sbm1_recovered.txt U | raven_hfn1_recovered.txt U

Homework8 > raven_sbm1_recovered.txt

```
THE RAVEN by Edgar Allan Poe

Once upon a midnight dreary, while I pondered, weak and weary,
Over many a quaint and curious volume of forgotten lore,
While I nodded, nearly napping, suddenly there came a tapping,
As of some one gently rapping, rapping at my chamber door.
"'Tis some visitor," I muttered, "tapping at my chamber door-
                Only this, and nothing more."

Ah, distinctly I remember it was in the bleak December,
And each separate dying ember wrought its ghost upon the floor.
Eagerly I wished the morrow;- vainly I had sought to borrow
From my books surcease of sorrow- sorrow for the lost Lenore-
For the rare and radiant maiden whom the angels name Lenore-
                Nameless here for evermore.

And the silken, sad, uncertain rustling of each purple curtain
Thrilled me- filled me with fantastic terrors never felt before;
So that now, to still the beating of my heart, I stood repeating,
"'Tis some visitor entreating entrance at my chamber door-
Some late visitor entreating entrance at my chamber door;-
                This it is, and nothing more."

Presently my soul grew stronger; hesitating then no longer,
"Sir," said I, "or Madam, truly your forgiveness I implore;
But the fact is I was napping, and so gently you came rapping,
And so faintly you came tapping, tapping at my chamber door,
That I scarce was sure I heard you"- here I opened wide the door;-
                Darkness there, and nothing more.

Deep into that darkness peering, long I stood there wondering, fearing,
Doubting, dreaming dreams no mortal ever dared to dream before;
But the silence was unbroken, and the stillness gave no token,
And the only word there spoken was the whispered word, "Lenore?"
This I whispered, and an echo murmured back the word, "Lenore!"-
                Merely this, and nothing more.

Back into the chamber turning, all my soul within me burning,
Soon again I heard a tapping somewhat louder than before.
"Surely," said I, "surely that is something at my window lattice:
Let me see, then, what thereat is, and this mystery explore-
Let my heart be still a moment and this mystery explore;-
```

main* 0 0 Ln 1, Col 1 Spaces: 4 UTF-8 LF {} Plain Text Go Live

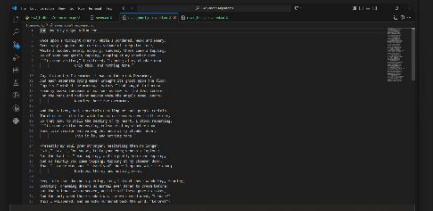
File Edit Selection View Go Run Terminal Help

2.Advanced Algorithms

Hw8_HuffmanCompression.py U | raven.txt U | raven_sbm1_recovered.txt U | raven_hfn1_recovered.txt U

Homework8 > raven_hfn1_recovered.txt

```
THE RAVEN by Edgar Allan Poe

Once upon a midnight dreary, while I pondered, weak and weary,
Over many a quaint and curious volume of forgotten lore,
While I nodded, nearly napping, suddenly there came a tapping,
As of some one gently rapping, rapping at my chamber door.
"'Tis some visitor," I muttered, "tapping at my chamber door-
                Only this, and nothing more."

Ah, distinctly I remember it was in the bleak December,
And each separate dying ember wrought its ghost upon the floor.
Eagerly I wished the morrow;- vainly I had sought to borrow
From my books surcease of sorrow- sorrow for the lost Lenore-
For the rare and radiant maiden whom the angels name Lenore-
                Nameless here for evermore.

And the silken, sad, uncertain rustling of each purple curtain
Thrilled me- filled me with fantastic terrors never felt before;
So that now, to still the beating of my heart, I stood repeating,
"'Tis some visitor entreating entrance at my chamber door-
Some late visitor entreating entrance at my chamber door;-
                This it is, and nothing more."

Presently my soul grew stronger; hesitating then no longer,
"Sir," said I, "or Madam, truly your forgiveness I implore;
But the fact is I was napping, and so gently you came rapping,
And so faintly you came tapping, tapping at my chamber door,
That I scarce was sure I heard you"- here I opened wide the door;-
                Darkness there, and nothing more.

Deep into that darkness peering, long I stood there wondering, fearing,
Doubting, dreaming dreams no mortal ever dared to dream before;
But the silence was unbroken, and the stillness gave no token,
And the only word there spoken was the whispered word, "Lenore?"
This I whispered, and an echo murmured back the word, "Lenore!"-
                Merely this, and nothing more.

Back into the chamber turning, all my soul within me burning,
Soon again I heard a tapping somewhat louder than before.
"Surely," said I, "surely that is something at my window lattice:
Let me see, then, what thereat is, and this mystery explore-
Let my heart be still a moment and this mystery explore;-
```



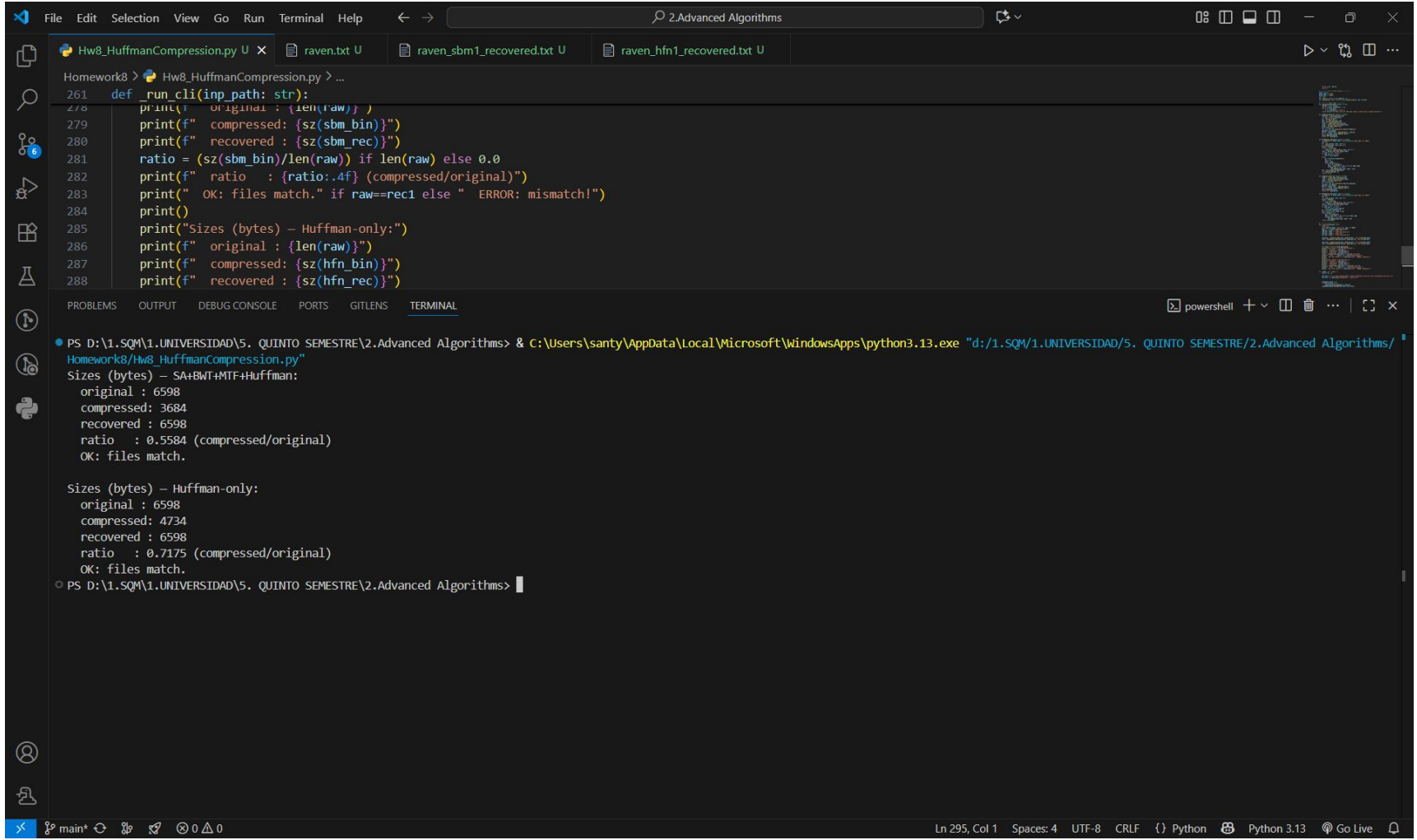
Herramienta Recortes

Captura de pantalla copiada en el portapapeles
Guardado automáticamente en la carpeta de capturas de pantalla.

Marcado y uso compartido

main* 0 0 0

Ln 1, Col 1 Spaces: 4 UTF-8 LF Plain Text Go Live


File Edit Selection View Go Run Terminal Help
2.Advanced Algorithms
Hw8_HuffmanCompression.py U
raven.txt U
raven_sbm1_recovered.txt U
raven_hfn1_recovered.txt U
Homework8 > Hw8_HuffmanCompression.py > ...
261 def _run_cli(inp_path: str):
279 print(f" compressed: {sz(sbm_bin)}")
280 print(f" recovered : {sz(sbm_rec)}")
281 ratio = (sz(sbm_bin)/len(raw)) if len(raw) else 0.0
282 print(f" ratio : {ratio:.4f} (compressed/original)")
283 print(" OK: files match." if raw==rec1 else " ERROR: mismatch!")
284 print()
285 print("Sizes (bytes) — Huffman-only:")
286 print(f" original : {len(raw)}")
287 print(f" compressed: {sz(hfn_bin)}")
288 print(f" recovered : {sz(hfn_rec)}")
PROBLEMS OUTPUT DEBUG CONSOLE PORTS GITLENS TERMINAL
powershell
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms> & C:\Users\santy\AppData\Local\Microsoft\WindowsApps\python3.13.exe "d:/1.SQM/1.UNIVERSIDAD/5. QUINTO SEMESTRE/2.Advanced Algorithms/Homework8/Hw8_HuffmanCompression.py"
Sizes (bytes) — SA+BWT+MTF+Huffman:
original : 6598
compressed: 3684
recovered : 6598
ratio : 0.5584 (compressed/original)
OK: files match.
Sizes (bytes) — Huffman-only:
original : 6598
compressed: 4734
recovered : 6598
ratio : 0.7175 (compressed/original)
OK: files match.
PS D:\1.SQM\1.UNIVERSIDAD\5. QUINTO SEMESTRE\2.Advanced Algorithms>
main*
Ln 295, Col 1 Spaces: 4 UTF-8 CRLF Python Python 3.13 Go Live

https://colab.research.google.com/drive/1jYTL1eGGLmnzfKvl-6MLFra4Y0Q3gwY8?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

GeeksforGeeks. (2025, July 23). *Suffix Array | Set 1 (Introduction)*. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/suffix-array-set-1-introduction/

GeeksforGeeks. (2025, August 14). *Burrows Wheeler Data Transform Algorithm*. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/burrows-wheeler-data-transform-algorithm/

GeeksforGeeks. (2023, March 29). *Move to front data transform algorithm*. GeeksforGeeks. https://www.geeksforgeeks.org/dsa/move-front-data-transform-algorithm/