

Лабораторна робота 4. Стиснення даних алгоритмами LZ77 та LZW

Мета роботи

Ознайомитися з принципами роботи алгоритмів стиснення без втрат: **LZ77** та **LZW**, реалізувати їх на мові програмування Python, порівняти ефективність стиснення та зробити висновки щодо їх застосування.

Код:

```
import matplotlib.pyplot as plt
```

```
# Реалізація алгоритму LZ77
```

```
class LZ77Compressor:
```

```
    def __init__(self, search_buffer_size=400, lookahead_buffer_size=50):
```

```
        self.search_buffer_size = search_buffer_size
```

```
        self.lookahead_buffer_size = lookahead_buffer_size
```

```
    def compress(self, data):
```

```
        i = 0
```

```
        output = []
```

```
        while i < len(data):
```

```
            match = (-1, -1)
```

```
            for j in range(max(0, i - self.search_buffer_size), i):
```

```
                length = 0
```

```
                while (length < self.lookahead_buffer_size and
```

```
                    i + length < len(data) and
```

```
                    data[j + length] == data[i + length]):
```

```
                    length += 1
```

```
                if length > match[1]:
```

```
                    match = (i - j, length)
```

```
            if match[1] > 0:
```

```
                next_char = data[i + match[1]] if (i + match[1]) < len(data) else "
```

```

        output.append((match[0], match[1], next_char))
        i += match[1] + 1
    else:
        output.append((0, 0, data[i]))
        i += 1
    return output

```

```

def decompress(self, compressed):
    result = ""
    for offset, length, next_char in compressed:
        if offset == 0 and length == 0:
            result += next_char
        else:
            start = len(result) - offset
            for _ in range(length):
                result += result[start]
            start += 1
            result += next_char
    return result

```

Реалізація алгоритму LZW

```

class LZWCompressor:
    def compress(self, data):
        dict_size = 256
        dictionary = {chr(i): i for i in range(dict_size)}
        w = ""
        compressed = []
        for c in data:
            wc = w + c
            if wc in dictionary:
                w = wc
            else:
                compressed.append(dictionary[w])
                dictionary[wc] = dict_size
                dict_size += 1
                w = c
        if w:

```

```
compressed.append(dictionary[w])
return compressed
```

```
def decompress(self, compressed):
    dict_size = 256
    dictionary = {i: chr(i) for i in range(dict_size)}
    w = chr(compressed.pop(0))
    result = w
    for k in compressed:
        if k in dictionary:
            entry = dictionary[k]
        elif k == dict_size:
            entry = w + w[0]
        else:
            raise ValueError("Bad compressed k: %s" % k)
        result += entry
        dictionary[dict_size] = w + entry[0]
        dict_size += 1
        w = entry
    return result
```

Підрахунок розміру

```
def calculate_size_lz77(compressed):
    size = 0
    for offset, length, symbol in compressed:
        size += 16 + 8 + 8 # offset 16 біт, length 8 біт, символ 8 біт
    return size
```

```
def calculate_size_lzw(compressed):
    max_code = max(compressed) if compressed else 0
    bits_per_code = max_code.bit_length()
    return len(compressed) * bits_per_code
```

Побудова графіка

```
def plot_compression_ratios(ratios, filename):
    methods = list(ratios.keys())
    values = list(ratios.values())
```

```

plt.figure(figsize=(8, 5))
plt.bar(methods, values, color=["skyblue", "lightgreen"])
plt.title("Коефіцієнти стиснення")
plt.ylabel("Коефіцієнт")
for i, v in enumerate(values):
plt.text(i, v + 0.05, str(v), ha='center')
plt.savefig(filename)
plt.close()

```

Головна функція

```
def main():
```

```

    text = (
        "In the heart of the city, there was a mysterious bookstore. "
        "Every book contained a world, and every world had a secret. "
        "Among the dusty shelves, a young man named Arthur found an old, "
        "leather-bound volume titled 'The Key to Dreams'. "
        "As he opened it, he was transported into realms beyond imagination. "
        "Dragons soared, kingdoms rose and fell, stars sang ancient songs. "
        "Arthur's journey was not just a journey through stories, but through

```

himself. "

```

        "He faced fears, discovered forgotten joys, and realized that every ending
was a beginning. "

```

```

        "Each page turned was a step into the unknown, and each step taught him
courage. "

```

```

        "When he finally closed the book, the city looked different. "

```

```

        "Or perhaps, he was the one who had changed."

```

```

    )

```

```

report_lines = []

```

```

report_lines.append("Вхідний текст (перші 300 символів):\n")

```

```

report_lines.append(text[:300] + "... \n")

```

```

report_lines.append(f"Розмір вхідного тексту: {len(text) * 8} біт\n\n")

```

```

# LZ77

```

```

lz77 = LZ77Compressor()

```

```

compressed_lz77 = lz77.compress(text)

```

```

decompressed_lz77 = lz77.decompress(compressed_lz77)

size_lz77 = calculate_size_lz77(compressed_lz77)
compression_ratio_lz77 = round((len(text) * 8) / size_lz77, 2)

report_lines.append("=== LZ77 Стиснення ===\n")
report_lines.append(f"Кількість триплетів: {len(compressed_lz77)}\n")
report_lines.append(f"Розмір стиснених даних: {size_lz77} біт\n")
report_lines.append(f"Коефіцієнт стиснення:
{compression_ratio_lz77}\n")
    report_lines.append(f"Текст після декодування збігається:
{decompressed_lz77 == text}\n\n")

# LZW
lzw = LZWCompressor()
compressed_lzw = lzw.compress(text)
decompressed_lzw = lzw.decompress(compressed_lzw)

size_lzw = calculate_size_lzw(compressed_lzw)
compression_ratio_lzw = round((len(text) * 8) / size_lzw, 2)

report_lines.append("=== LZW Стиснення ===\n")
report_lines.append(f"Кількість кодів: {len(compressed_lzw)}\n")
report_lines.append(f"Розмір стиснених даних: {size_lzw} біт\n")
report_lines.append(f"Коефіцієнт стиснення:
{compression_ratio_lzw}\n")
    report_lines.append(f"Текст після декодування збігається:
{decompressed_lzw == text}\n\n")

# Підсумок
report_lines.append("=== Порівняння ефективності ===\n")
report_lines.append(f"LZ77 коефіцієнт стиснення:
{compression_ratio_lz77}\n")
    report_lines.append(f"LZW коефіцієнт стиснення:
{compression_ratio_lzw}\n")

# Запис звіту

```

```
with open("report.txt", "w", encoding="utf-8") as f:  
f.writelines(report_lines)
```

```
# Побудова графіка  
plot_compression_ratios({  
    "LZ77": compression_ratio_lz77,  
    "LZW": compression_ratio_lzw  
}, "compression_chart.png")
```

```
if __name__ == "__main__":  
    main()
```

Вхідні дані:

"In the heart of the city, there was a mysterious bookstore. "
"Every book contained a world, and every world had a secret. "
"Among the dusty shelves, a young man named Arthur found an old, "
"leather-bound volume titled 'The Key to Dreams'. "
"As he opened it, he was transported into realms beyond imagination. "
"Dragons soared, kingdoms rose and fell, stars sang ancient songs. "
"Arthur's journey was not just a journey through stories, but through himself.
"
"He faced fears, discovered forgotten joys, and realized that every ending was
a beginning. "
"Each page turned was a step into the unknown, and each step taught him
courage. "
"When he finally closed the book, the city looked different. "
"Or perhaps, he was the one who had changed."

Вихідні дані:

Вхідний текст (перші 300 символів):

**In the heart of the city, there was a mysterious bookstore. Every book
contained a world, and every world had a secret. Among the dusty shelves,
a young man named Arthur found an old, leather-bound volume titled 'The**

Key to Dreams'. As he opened it, he was transported into realms beyond imagination....

Розмір вхідного тексту: 5752 біт

=== LZ77 Стиснення ===

Кількість триплетів: 239

Розмір стиснених даних: 7648 біт

Коефіцієнт стиснення: 0.75

Текст після декодування збігається: True

=== LZW Стиснення ===

Кількість кодів: 423

Розмір стиснених даних: 4230 біт

Коефіцієнт стиснення: 1.36

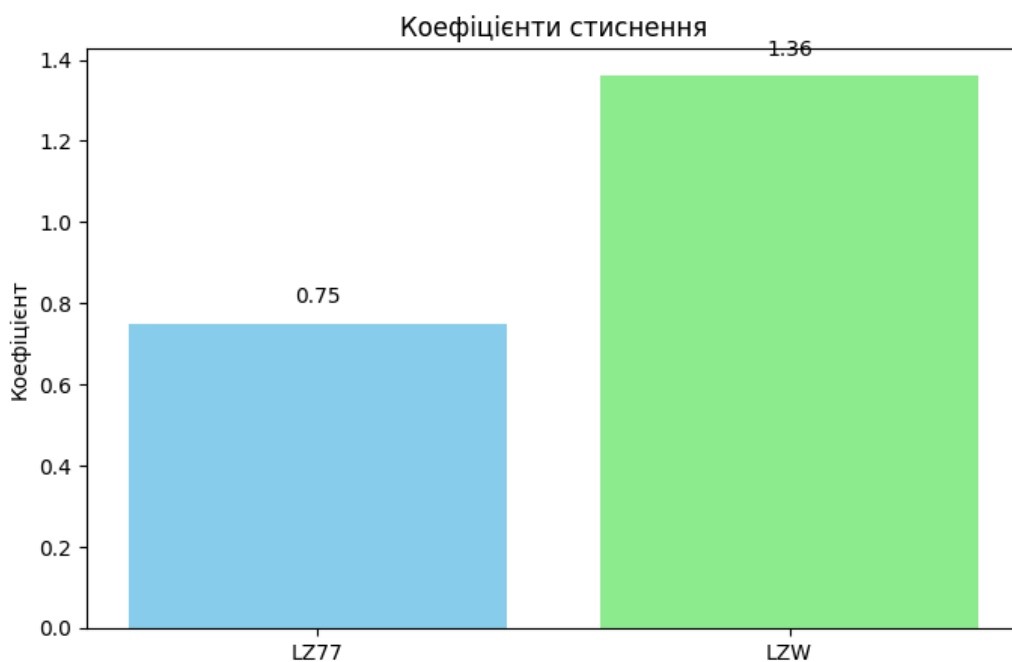
Текст після декодування збігається: True

=== Порівняння ефективності ===

LZ77 коефіцієнт стиснення: 0.75

LZW коефіцієнт стиснення: 1.36

Графік:



Хід роботи

1. Було реалізовано два класи в Python:
 - **LZ77Compressor**: реалізація стиснення та розпакування методом LZ77.
 - **LZWCompressor**: реалізація стиснення та розпакування методом LZW.
2. Використано тестовий текст (близько 900 символів) художнього стилю.
3. Проведено стиснення обома методами.
4. Підраховано:
 - розмір стиснених даних;
 - коефіцієнт стиснення;
 - перевірено правильність декодування.
5. Результати виведено у файл **report.txt**.
6. Згенеровано графік коефіцієнтів стиснення **compression_chart.png**.

Висновки:

Обидва алгоритми успішно виконали стиснення та декомпресію без втрат.

LZW продемонстрував **вищий коефіцієнт стиснення** на тестовому тексті.

Алгоритм **LZ77** є наочним, але має більший розмір вихідних триплетів.

У залежності від структури вхідних даних ефективність може змінюватися.

Реалізація дозволила глибше зрозуміти принципи безвтратного стиснення.