Jakub Janicki

Sprawozdanie z laboratorium 3 – kodowanie Huffmana

1. Zostały zaimplementowane 2 algorytmy kompresji oraz dekompresji:
   1.1. statyczny algorytm Huffmana
   1.2. dynamiczny algorytm Huffmana

2. Zostały przygotowane następujące pliki testowe:
   2.1. 1KB.txt
   2.2. 10KB.txt
   2.3. 100KB.txt
   2.4. 1MB.txt
   Powyższe pliki uzyskano komendą *cat /dev/urandom | tr -dc 'a-zA-Z' | fold -w R* | head -1 > r*.txt*
   Gdzie R to rozmiar pliku w bytach, a r to nazwa pliku
   2.5. guttenberg.txt – tekst literacki (671 KB)
   2.6. linux.txt – plik źródłowy Linux (650 KB)
   2.7. uniform.txt – plik ze znakami losowanymi z rozkładu jednostajnego, utworzony za pomocą random.uniform (650 KB)

3. Została sprawdzona poprawność działania algorytmów na plikach testowych:

```
Static huffman   passed 1KB.txt test
Adaptive huffman passed 1KB.txt test
Static huffman   passed 10KB.txt test
Adaptive huffman passed 10KB.txt test
Static huffman   passed 100KB.txt test
Adaptive huffman passed 100KB.txt test
Static huffman   passed 1MB.txt test
Adaptive huffman passed 1MB.txt test
Static huffman   passed guttenberg.txt test
Adaptive huffman passed guttenberg.txt test
Static huffman   passed linux.txt test
Adaptive huffman passed linux.txt test
Static huffman   passed uniform.txt test
Adaptive huffman passed uniform.txt test
```

4. Zostały dokonane pomiary czasowe działania algorytmów
   4.1. Czasy kompresji

ENCODE	1KB.txt	10KB.txt	100KB.txt	1MB.txt	guttenberg.txt	linux.txt	uniform.txt
Static Huffman	0.001	0.003	0.029	0.349	0.187	0.19	0.258
Adaptive Huffman	0.033	0.243	2.334	23.893	12.517	13.667	15.484
Static - Adaptive	-0.032	-0.24	-2.305	-23.544	-12.33	-13.477	-15.226

   4.2. Czasy dekompresji

DECODE	1KB.txt	10KB.txt	100KB.txt	1MB.txt	guttenberg.txt	linux.txt	uniform.txt
Static Huffman	0.001	0.018	0.127	1.433	0.829	0.847	0.938
Adaptive Huffman	0.023	0.249	2.322	26.907	12.372	14.758	15.707
Static - Adaptive	-0.022	-0.231	-2.195	-25.474	-11.543	-13.911	-14.769

4.3. Kompresja

COMPRESSION	1KB.txt	10KB.txt	100KB.txt	1MB.txt	guttenberg.txt	linux.txt	uniform.txt
Static Huffman	28.42%	27.87%	27.72%	27.66%	43.63%	28.83%	27.91%
Adaptive Huffman	22.36%	27.09%	27.59%	27.64%	44.16%	32.32%	27.63%
Static - Adaptive	6.06%	0.78%	0.13%	0.02%	-0.53%	-3.49%	0.28%

Wnioski:

1. Pliki testowe pozwoliły przetestować program pod każdym względem. Plik guttenberg.txt sprawdził algorytmy pod kątem radzenia sobie z nietypowymi znakami. Plik linux.txt zawierający ‘#’ zmusił do zmiany znaku NYC z ‘#’ na prywatny unicode "\U000F0000". Pliki o różnej długości oraz pliki o tej samej długości i różnej statystyce występowania znaków pozwalają na analizę czasową działania algorytmów.
2. Zarówno czasu kompresji i dekompresji są większe dla algorytmu dynamicznego. Czasy działania algorytmów zwiększają się wraz z rozmiarem tekstu. Dla algorytmu dynamicznego względny wzrost czasu do wzrostu rozmiaru jest dużo większy niż dla algorytmu statycznego, przez co największe różnice w czasie możemy zauważyć dla większych plików np. dla 1MB.txt to 23,5s.
3. Czasy dekompresji są większe niż czasy kompresji dla algorytmu statycznego. Dla dynamicznego są porównywalne.
4. Kompresja jest porównywalna dla obu algorytmów. Zgodnie z oczekiwaniami, najgorzej skompresowane zostały pliki o jednostajnym rozkładzie ok 27%. Najlepiej został skompresowany plik guttenberg.txt aż 44%.