Ann Mai
Cody Sardella
CS 165
Project 3

Lempel-Ziv Sliding Window

I. Compilation Instructions

Use -std=c++11 flag to compile both LZ and EXPAND source files.

- To compile the compression program, navigate to the source folder and run the command:
  g++ -std=c++11 mainLZ.cpp LZ.cpp -o LZ

- To compile the expand program navigate to the same source folder and run the command:
  g++ -std=c++11 mainEXP.cpp EXPAND.cpp -o EXPAND

II. Execution

- To compress a file:

  Run the command with 0 to 3 flags and a required filename of a file in the directory:
  LZ [flags] [filename]

  FLAGS:

  -N : Window Offset Size
  Min Value: 9 ….. Max Value: 14
  -L : Match Length
  Min Value: 3 ….. Max Value: 4
  -S: String Encoding Size
  Min Value: 1 ….. Max Value: 5

- To expand a file:

  Run the command with an optional file name: EXPAND [filename]

III. Data Structures Used

- string data type was used to construct the Lempel-Ziv window
- vector of chars was used to store the encoded binary data
- Standard library map<string, int> was used to implement a dictionary of prefixes and the corresponding offsets

IV. Analysis

LZ:
In the basics of this algorithm, it tries to find the longest match using the prefix of the lookahead buffer. After the longest match has been found, the match gets encoded, and the window shifts forward the length of the longest match.

N = number of bits to encode window offset
L = number of bits to encode maximum match length
S = number of bits to encode maximum literal string length
d = size of dictionary
2^N = window size
2^L = lookahead buffer size
2^S-1 = longest encoded literal size
α = non-constant smaller than 2^L that equals average longest match length
β = non-constant smaller than 2^S-1 that equals average literal string length

Basic Operations	**Worst Case**	**Average Case**
Search for longest matches	$2^L * (2^N - 2^L)$	$\alpha * (2^N - 2^L)$
Search in Dictionary	O(log d)	O(1)*
Insert in Dictionary	O(log d)	O(1)*
Remove from/Update Dictionary	O(d)	O(d)
Encode/Decode Token Double	L + N	L + N
Encode/Decode Token Triple	$L + S + (2^S - 1) * 8$	$L + S + 8\beta$

* For any prefix substring that may already exist in the dictionary, the match is found in O(log d) time, where d is the size of the dictionary. d remains relatively small because as the window slides, prefixes of substrings outside of the window get removed. This makes lookup and insertion closer to O(1).

V. Optimal LZ Parameter Values

- kennedy.xls N=9, L=4, S=1 3.57 seconds 71% compression savings
- book1 N=13, L=3, S=5 62.57 seconds 54% compressions savings

VI. Recorded Results

Parameters			kennedy.xls		book1	
N	L	S	Time (sec)	Compression Savings %	Time (sec)	Compression Savings %
9	3	1	4.00	63	8.21	34
9	3	2	4.01	63	8.27	35
9	3	3	4.02	62	8.39	34
9	3	4	4.03	61	8.19	33
9	3	5	4.04	60	8.21	32
9	**4**	**1**	**3.57**	**71**	8.23	28
9	4	2	3.63	70	8.20	30
9	4	3	3.62	69	8.39	29
9	4	4	3.62	68	8.29	28
9	4	5	3.65	67	8.30	27
N	L	S	Time (sec)	Compression Savings %		
10	3	1	6.47	62	13.34	40
10	3	2	6.34	61	13.37	40
10	3	3	6.34	60	13.34	40
10	3	4	6.36	59	13.28	39
10	3	5	6.39	58	13.35	39
10	4	1	5.66	70	13.38	36
10	4	2	5.62	70	13.30	36
10	4	3	5.66	69	13.37	36
10	4	4	5.65	68	13.37	35
10	4	5	5.70	67	13.41	34
N	L	S	Time (sec)	Compression Savings %		
11	3	1	10.86	60	22.03	45
11	3	2	10.87	59	21.89	45
11	3	3	10.87	59	22.03	44
11	3	4	10.89	58	21.87	44
11	3	5	10.93	57	22.83	44
11	4	1	9.53	70	22.14	41
11	4	2	9.57	69	21.91	41
11	4	3	9.57	68	22.09	41
11	4	4	9.61	67	22.10	41
11	4	5	9.59	66	21.89	40
N	L	S	Time (sec)	Compression Savings %		
12	3	1	25.92	70	36.84	48
12	3	2	25.94	70	36.81	48
12	3	3	25.73	70	36.57	48
12	3	4	25.94	70	36.79	48
12	3	5	25.72	70	36.81	48
12	4	1	29.14	68	36.51	46
12	4	2	29.20	68	36.80	46
12	4	3	29.18	68	36.80	45
12	4	4	29.16	68	36.85	45
12	4	5	29.15	68	36.78	45

Parameters			kennedy.xls		book1	
N	L	S	Time (sec)	Compression Savings %	Time (sec)	Compression Savings %
13	3	1	42.99	68	62.55	51
13	3	2	42.95	68	62.60	51
13	3	3	42.97	68	62.03	51
13	3	4	42.93	68	62.47	51
13	**3**	**5**	42.95	68	**62.57**	**54**
13	4	1	47.11	66	61.99	49
13	4	2	47.08	66	62.55	49
13	4	3	47.09	66	62.55	49
13	4	4	46.64	66	62.02	49
13	4	5	47.10	66	62.53	49
N	L	S	Time (sec)	Compression Savings %	Time (sec)	Compression Savings %
14	3	1	68.84	66	107.66	53
14	3	2	68.75	66	106.77	53
14	3	3	68.77	66	107.57	53
14	3	4	68.81	66	107.55	53
14	3	5	68.56	66	106.64	53
14	4	1	73.49	64	107.45	52
14	4	2	71.86	64	107.34	52
14	4	3	72.18	64	107.45	52
14	4	4	72.13	64	106.48	52
14	4	5	72.12	64	107.37	52