

I. GRUPAS APRAKSTS

Grupas nosaukums

POGramētāji

Grupas locekļi

- Tomass Kristiāns Šterns, 5.grupa, 231RDB190
- Petr Gabuniia, 6.grupa, 231RDB331
- Valentīns Koposovs, 10.grupa, 231RDB008

II. PIRMAIS TESTA PIEMĒRS

Oriģinālfaila teksts tika ņemts no https://undertale.fandom.com/wiki/Temmie_Armor, jo bija vēlme testa failam paņemt kaut ko interesantu, kādu eksistējošu un lietojamu teksta gabalu un šis bija viens no pirmajiem, kas pēc nejaušības iešāvās prātā. Faila izmērs: 100 baiti.

Sample.txt ×	
54 65 6D 6D 69 65 20 41	72 6D 6F 72 20 69 73 20 Temmie Armor is
61 6E 20 61 72 6D 6F 72	20 69 74 65 6D 20 73 6F an armor item so
6C 64 20 61 74 20 74 68	65 20 54 65 6D 20 53 68 ld at the Tem Sh
6F 70 2E 20 49 74 20 69	73 20 74 68 65 20 73 65 op. It is the se
63 6F 6E 64 20 6D 6F 73	74 20 70 6F 77 65 72 66 cond most powerf
75 6C 20 61 72 6D 6F 72	20 69 6E 20 74 68 65 20 ul armor in the
67 61 6D 65	game

Tad šis fails tika apstrādāts ar LZ77 algoritmu, kura meklēšanas un priekšskatīšanas bufera izmēri bija viss fails un, tā kā faila izmērs ir 100 baitu, tad var pieņemt ka bufera izmēri bija 100 baiti katrs. Pēc LZ77 saspiešanas faila saturs tika aizvietots ar baitu trijniekiem – distance, daudzums, nākamais simbols –, kur skaitliskās vērtības tika pārvērstas par char, tādā veidā ietaupot vietu. Faila izmērs: 132 baiti.

LZ77_Sample.txt ✕																	
00000000	00	00	54	00	00	65	00	00	6D	01	01	69	04	01	20	00	..T..e..m..i..
00000010	00	41	00	00	72	07	01	6F	03	01	20	09	01	73	09	01	.A..r..o.. ..s..
00000020	61	00	00	6E	03	02	72	0C	05	74	1A	02	20	10	01	6F	a..n..r..t.. ..o
00000030	00	00	6C	00	00	64	13	02	74	1F	01	74	00	00	68	23	..l..d..t..t..h#
00000040	02	54	10	03	53	08	01	6F	00	00	70	00	00	2E	2D	01	.T..S..o..p...-
00000050	49	11	02	69	2A	02	74	14	03	73	3E	01	63	37	01	6E	I..i*.t..s>.c7.n
00000060	22	02	6D	3C	01	73	24	02	70	41	01	77	4C	01	72	00	".m<.s\$.pA.wL.r.
00000070	00	66	00	00	75	31	01	20	40	07	6E	36	05	67	51	01	.f..ul. @.n6.gQ.
00000080	6D	62	00	65											mb.e		

Visbeidzot, tika pielietots Huffman saspiešanas algoritms, kas saskaitīja visu simbolu biežumu, izveidoja biežumu tabulu, pēc kuras tika izveidots koks, no kura tika izlasīti simboli un tiem atbilstošās vērtības, kuras tika ierakstītas faila sākumā kā simbols, bitu daudzums un pati bitu virkne, kuri tika pārveidoti un ierakstīti kā char, lai ietaupītu vietu, un pārējais faila saturs ir atbilstošu simbolu aizvietošana ar bitu virknēm, kuras pēc tam salika vienu otrai priekšā un izveidojās gala saspiešanas fails. Faila izmērs: 288 baiti.

Huffman_LZ77_Samp... x																	
00000000	00	02	00	01	03	06	02	04	08	03	05	09	20	05	0D	74t
00000010	05	17	6D	06	3A	6E	06	3C	6F	06	3D	72	06	3B	73	06	..m.:n.<o.=r.;s.
00000020	3E	05	06	28	07	06	11	09	06	2C	10	06	24	41	07	7E	>..(.....,\$A.~
00000030	54	06	15	65	06	18	69	06	2B	70	07	7F	04	07	20	08	T..e..i.+p.△..
00000040	07	29	0C	07	4F	11	07	54	13	07	55	14	07	39	1A	07	.)..0..T..U..9..
00000050	3B	1F	07	52	22	07	53	23	07	32	24	07	33	2A	07	3C	;..R".S#.2\$.3*.<
00000060	2D	07	3D	2E	07	3F	31	07	2E	36	07	2F	37	07	70	3C	-.=..?1..6./7.p<
00000070	07	71	3E	07	4E	40	07	5B	49	07	4A	4C	07	4B	51	07	.q>.N@.[I.JL.KQ.
00000080	2C	53	07	2D	61	07	72	62	07	73	63	07	5A	64	07	4C	,S.-a.rb.sc.Zd.L
00000090	66	07	4D	67	07	38	68	07	3E	6C	07	3A	75	07	28	77	f.Mg.8h.>l.:u.(w
000000A0	07	21	05	41	C2	80	C3	AB	6A	C3	90	66	C2	87	C3	A0	!.ATÇ ½j ÉfTç á
000000B0	C3	AD	1D	C3	AA	73	6C	C3	9F	59	C2	B9	07	C2	89	C2	i. s fY T .TēT
000000C0	8E	C3	A7	C3	91	77	70	C3	99	37	C2	A0	C3	A8	26	55	Ä °æwp Ö7Tá z&U
000000D0	C2	8B	C3	92	C3	97	07	C3	8C	C2	A1	59	12	C2	B5	4E	T Æ ù. îTîY.T N
000000E0	C3	B4	3F	C2	83	C3	B7	C2	BA	55	48	C2	AD	C3	A4	5D	H?Tâ T UH T ñ]
000000F0	C3	8A	7E	C2	9D	C2	AD	70	C3	9E	53	C2	8E	C2	B8	C3	è~YTi p PSÄT
00000100	AF	C2	99	C3	87	C3	BF	C3	99	0C	C2	BD	C3	98	4D	05	»TÖ ç T Ö.T ÿM.
00000110	0B	C2	B3	6D	C2	A3	C3	A2	C3	BA	1C	2C	C3	9D	73	18	.T mTú ó H ., ¥s.
00000120																	

Šis fails vizuāli ir pamanāmi lielāks par oriģinālfailu, tas ir tāpēc, ka uz maziem faila izmēriem, Huffman-a simbolu-bitu atslēgas aizņem salīdzinošu faila daļu, kādēļ arī izmēri palielinās. Tomēr, lielākiem failiem, apvienojot LZ77 un Huffman, būtu jāsanāk samazināts faila izmērs.

Šis fails lielāks par oriģinālfailu, tas ir tāpēc, ka uz maziem faila izmēriem, Huffman-a simbolu-bitu atslēgas aizņem salīdzinošu faila daļu, kādēļ arī izmēri palielinās. Tomēr, lielākiem failiem, apvienojot LZ77 un Huffman, būtu jāsanāk samazināts faila izmērs. Kā arī pašu faila izmēru var ietekmēt pēc LZ77 saspiešanas esošo dažādu simbolu skaits, kas pie vairāk simboliem palielinās faila izmēru.

IV. TREŠAIS TESTA PIEMĒRS

Oriģinālfaila teksts tika izvēlēts, attēlojot C++ valodas standarta izvadi, kur tiek izvadīts teksts “Hello, World!”, kas mēdz būt pirmais teksts, kas tiek izvadīts, apgūstot jaunu programmēšanas valodu. Faila izmērs: 90 baiti.

```
Hello_World.txt x
00000000 23 69 6E 63 6C 75 64 65 20 3C 69 6F 73 74 72 65 #include <iostre
00000010 61 6D 3E 0D 0A 0D 0A 69 6E 74 20 6D 61 69 6E 28 am>....int main(
00000020 29 20 0D 0A 7B 0D 0A 20 20 20 20 73 74 64 3A 3A ) ..{.. std::
00000030 63 6F 75 74 20 3C 3C 20 22 48 65 6C 6C 6F 2C 20 cout << "Hello,
00000040 57 6F 72 6C 64 21 22 3B 0D 0A 20 20 20 20 72 65 World!";.. re
00000050 74 75 72 6E 20 30 3B 0D 0A 7D + turn 0;..}
```

Fails tika saspiegts, izmantojot LZ77 algoritmu. Tā saturs tika aizvietots ar baitu trijniekiem – distance, daudzums, nākamais simbols –, kur skaitliskās vērtības tika pārvērstas par char, tādā veidā ietaupot vietu. Faila izmērs: 156 baiti.

```
Hello_World.txt x LZ77_Hello_World.txt x
00000000 00 00 23 00 00 69 00 00 6E 00 00 63 00 00 6C 00 .#.i..n..c..l.
00000010 00 75 00 00 64 00 00 65 00 00 20 00 00 3C 09 01 .u..d..e.. ..<..
00000020 6F 00 00 73 00 00 74 00 00 72 08 01 61 00 00 6D o..s..t..r..a..m
00000030 00 00 3E 00 00 0D 00 00 0A 02 02 69 16 01 74 12 ..>.....i..t.
00000040 01 6D 0C 01 69 1C 01 28 00 00 29 19 01 0D 0F 01 .m..i..(..).....
00000050 7B 12 02 20 20 01 20 22 01 73 1F 01 64 00 00 3A {.. . "s..d..:
00000060 01 01 63 26 01 75 1A 02 3C 2D 01 20 00 00 22 00 ..c&u..<~.. ".
00000070 00 48 33 01 6C 38 01 6F 00 00 2C 37 01 57 36 01 .H3.l8.o..,7.W6.
00000080 72 3F 01 64 00 00 21 0E 01 3B 23 06 72 48 01 74 r?.d..!..;#.rH.t
00000090 4C 01 72 51 01 20 00 00 30 0F 03 7D + L.rQ. ..0..}
```

Visbeidzot, tika pieliets Huffman saspiešanas algoritms, kas saskaitīja visu simbolu biežumu, izveidoja biežumu tabulu, pēc kuras tika izveidots koks, no kura tika izlasīti simboli un tiem atbilstošās vērtības, un tika saglabāts failā kā char trijnieks, kas satur char, bitu skaits biežumu tabulā un bitu virkne biežumu tabulā, kā arī faila beigās tiek pievienots LZ77 šifrētā faila saturs pārveidojot to par bitu virkni attiecībā pret biežumu tabulu. Faila izmērs: 194 baiti.

```
FINAL_LZ77_Hello_W... x
00000000 3F 02 01 3F 03 04 64 05 16 72 05 15 23 05 04 63 ?..?...d...r...#...c
00000010 06 30 6C 06 3A 75 05 05 3C 06 33 6F 06 3F 73 06 .0l.:u...<.3o.?s.
00000020 2E 74 06 31 3F 06 34 3F 05 00 22 05 01 48 05 07 .t.1?.4?...".H..
00000030 30 07 64 33 06 0D 36 06 07 37 07 70 38 07 71 69 0.d3..6..7.p8.qi
00000040 07 78 6E 07 72 65 07 73 61 06 04 6D 06 05 3E 07 .xn.re.sa..m..>.
00000050 76 3F 07 77 7B 07 52 7D 07 53 3F 06 28 3F 07 7A v?.w{.R}.S?.(?z
00000060 3A 07 7B 26 07 6B 3F 07 5E 2D 07 7D 2C 07 6E 57 :.{&.k?.^-.},.nW
00000070 07 5F 3F 07 6F 21 07 6C 3F 07 7C 3B 07 6D 3F 07 ._?.o!.l?.|;.m?.
00000080 6A 4C 07 79 51 06 06 3F 06 0C 7D 07 65 27 3F 58 jL.yQ...?..}.e'?X
00000090 74 5B 73 CD FE EC 51 05 ED A3 4A 3F 21 6E F4 DB t[s=-∞Q.φúJ?!n[
000000A0 DA E1 AD 2D E0 67 F5 09 CD 7A E2 FF BB 3F 7C 76 rßi-αg].=ZΓ ȳ?|v
000000B0 BB DB 6C F8 ED 26 AA 3F C7 CB 51 3F 3F 19 4F 2F ȳl°φ&-?ȳQ??O/
000000C0 0E 00 + ..
```

Šis fails lielāks par oriģinālfailu, tas ir tāpēc, ka uz maziem faila izmēriem, Huffman-a simbolu-bitu atslēgas aizņem salīdzinošu faila daļu, kādēļ arī izmēri palielinās. Tomēr, lielākiem failiem, apvienojot LZ77 un Huffman, būtu jāsanāk samazināts faila izmērs. Kā arī pašu faila izmēru var ietekmēt pēc LZ77 saspiešanas esošo dažādu simbolu skaits, kas pie vairāk simboliem palielinās faila izmēru.