Serdica J. Computing **1** (2007), 13–26

Serdica
Journal of Computing

Bulgarian Academy of Sciences
Institute of Mathematics and Informatics

ON MULTIPLE DELETION CODES

Ivan Landjev, Kristiyan Haralambiev

Abstract. In 1965 Levenshtein introduced the deletion correcting codes and found an asymptotically optimal family of 1-deletion correcting codes. During the years there has been a little or no research on t-deletion correcting codes for larger values of t. In this paper, we consider the problem of finding the maximal cardinality $L_2(n,t)$ of a binary t-deletion correcting code of length n. We construct an infinite family of binary t-deletion correcting codes. By computer search, we construct t-deletion codes for $t = 2, 3, 4, 5$ with lengths $n \leq 30$. Some of these codes improve on earlier results by Hirschberg-Fereira and Swart-Fereira. Finally, we prove a recursive upper bound on $L_2(n,t)$ which is asymptotically worse than the best known bounds, but gives better estimates for small values of n.

1. Introduction. Let $F = \{0, 1, \ldots, q-1\}$ be a q-letter alphabet. A finite sequence of length n over F is called a q-ary word of length n. The set of all words of length n is denoted by F^n. A *block code* of length n over F is any subset C of F^n.

ACM Computing Classification System (1998): E.4.

Key words: insertion/deletion codes, Varshamov-Tennengolts codes, multiple insertion/deletion codes

Given $\boldsymbol{x} \in F^n$, we denote by $D_t(\boldsymbol{x})$ the set of all words from F^{n-t} obtained if any t letters are deleted from $\boldsymbol{x}$. In other words, $D_t(\boldsymbol{x})$ contains all subsequences of $\boldsymbol{x}$ of length $n-t$. Similarly, $I_t(\boldsymbol{x})$ denotes the set of all supersequences of length $n+t$, i.e. all words from F^{n+t} obtained if t letters are inserted in $\boldsymbol{x}$.

Definition 1.1. *The* Levenshtein distance $d_L(\boldsymbol{x}, \boldsymbol{y})$ *between two words* $\boldsymbol{x}, \boldsymbol{y}$ *from* F^n *is defined as the minimum number of deletions and insertions needed to transform* $\boldsymbol{x}$ *into* $\boldsymbol{y}$*.*

Clearly, $d(\boldsymbol{x}, \boldsymbol{y}) = 2(n - \ell(\boldsymbol{x}, \boldsymbol{y}))$, where $\ell(\boldsymbol{x}, \boldsymbol{y})$ is the length of the longest common subsequence of $\boldsymbol{x}$ and $\boldsymbol{y}$. Clearly$d_L(\boldsymbol{x}, \boldsymbol{y})$ is a metric on F^n.

Definition 1.2. *A code* $C \subseteq F^n$ *is called a* t-deletion correcting code *if* $D_t(\boldsymbol{x}) \cap D_t(\boldsymbol{y}) = \emptyset$ *for any* $\boldsymbol{x}, \boldsymbol{y} \in C$, $\boldsymbol{x} \neq \boldsymbol{y}$*.*

Definition 1.3. *A code* $C \subseteq F^n$ *is called a* t-insertion correcting code *if* $I_t(\boldsymbol{x}) \cap I_t(\boldsymbol{y}) = \emptyset$ *for any* $\boldsymbol{x}, \boldsymbol{y} \in C$, $\boldsymbol{x} \neq \boldsymbol{y}$*.*

Definition 1.4. *A code* $C \subseteq F^n$ *is called a* t-insertion/deletion correcting code *if* $d_L(\boldsymbol{x}, \boldsymbol{y}) > 2t$ *for any* $\boldsymbol{x}, \boldsymbol{y} \in C$, $\boldsymbol{x} \neq \boldsymbol{y}$*.*

It has been proved in [4] that t-deletion correcting codes, t-insertion correcting codes and t-insertion/deletion codes are essentially the same objects. In what follows we formulate all our results for t-deletion codes. A central problem about deletion codes is the following:

Given the integers n, t, $1 \leq t < n$, find the maximal cardinality $L_q(n,t)$ of a t-deletion correcting code over F.

A q-ary code C of length n correcting t deletions with $|C| = L_q(n,t)$ is called *optimal*. In this paper, we focus on *binary* t-deletion correcting codes. This is by far the most investigated family of deletion codes. The following asymptotic bounds have been proved by Levenshtein in [4].

Theorem 1.5. *For any fixed positive integer* t *and* $n \to \infty$

$$\frac{2^t (t!)^2 2^n}{n^{2t}} \lesssim L_2(n,t) \lesssim \frac{t! 2^n}{n^t},$$

where $f(n) \lesssim g(n)$ *means that* $\lim_{n\to\infty} f(n)/g(n) \leq 1$.

In the same paper, Levenshtein proved that $L_2(n,1) \geq \dfrac{2^n}{n+1}$ which implies that $L_2(n,1) \sim \dfrac{2^n}{n}$. He noticed that the so-called Varshamov-Tenengolts codes $VT_a(n)$ discovered in [13] are 1-deletion correcting codes. The Varshamov-Tenengolts code $VT_a(n)$, $0 \leq a \leq n$, consists of all binary vectors $(x_1, \ldots, x_n)$ satisfying

$$\sum_{i=1}^{n} i x_i \equiv a \pmod{n+1}.$$

The cardinality of these codes has been determined by Varshamov for $a = 0$ [12], by Ginzburg for $a = 1$ [1] and by Martirosyan [8] for any a (cf. also [9]).

Theorem 1.6.

$$|VT_a(n)| = \frac{1}{2n+1} \sum_{d|n,\ d\ odd} \phi(d) \frac{\mu\left(\frac{d}{(d,a)}\right)}{\phi\left(\frac{d}{(d,a)}\right)} 2^{(n+1)/d},$$

where ϕ is the Euler function, $\mu(n)$ is the Möbius function and (d,a) is the greatest common divisor of d and a.

This implies the following corollary.

Corollary 1.7 [9]**.**

(*i*) $|VT_0(n)| = \dfrac{1}{2(n+1)} \displaystyle\sum_{\substack{d|n+1 \\ d\ odd}} \phi(d) 2^{(n+1)/d}$;

(*ii*) $|VT_1(n)| = \dfrac{1}{2(n+1)} \displaystyle\sum_{\substack{d|n+1 \\ d\ odd}} \mu(d) 2^{(n+1)/d}$;

(*iii*) $|VT_0(n)| \geq |VT_a(n)| \geq |VT_1(a)|$.

The codes $VT_0(n)$ are optimal for all $n \leq 8$ [9] and very close to being optimal for large n. Due to the asymptotic estimate by Levenshtein (Theorem 1.5), one has $|VT_0(n)| \geq 2^n/(n+1)$, a result which is not transparent from Corollary 1.7. It is conjectured that $VT_0(n)$ are optimal for every n.

A code C is called *perfect* t-deletion correcting code if the balls $D_t(\boldsymbol{x})$, $\boldsymbol{x} \in C$, partition the set F^{n-t}. Remarkably, all the codes $VT_a(n)$, $a = 0, \ldots, n$, are perfect codes.

The question about the value of $L_2(n,t)$ for $t \geq 2$, i.e. about the maximal cardinalities of binary codes of length n correcting more than one deletion, is far less clear. Multiple deletion correcting codes were constructed in [2] (for $t = 2, 3, 4, 5,\ n \leq 14$) and in [10] (for $t = 2,\ n \leq 12$). The codes in the first paper are obtained in an attempted generalization of the Varshamov-Tenengolts codes. The codes in the second paper are obtained as a result of a greedy search performed on $5 \cdot 10^4$ random permutations of the 2^n binary words of length n. No infinite classes of multiple deletion correcting codes have been proposed so far.

The aim of this paper is to present constructions and upper bounds that improve on the results of [2] and [10] for codes that correct $t \geq 2$ deletions. In section 2, we give constructions for multiple deletion codes. We present the results of a computer search that was performed for $t = 2, 3, 4, 5$ and $n \leq 30$. Some of the constructed codes have larger cardinality than the largest codes known previously. In section 3, we start by proving some exact values for $L_2(n, t)$. Then we present a recursive upper bound which gives better estimates for small n than the best known upper bound on $L_2(n, t)$.

2. Constructions for t-deletion codes.

We start with a cascade construction for multiple deletion codes.

Theorem 2.1. *Let C be a t-deletion correcting code of length n. Then the code*

$$C^{(s)} = \{(\underbrace{c_1 \ldots c_1}_{s}, \underbrace{c_2 \ldots c_2}_{s}, \ldots, \underbrace{c_n \ldots c_n}_{s}) \mid (c_1, c_2, \ldots, c_n) \in C\}$$

is a code of length sn correcting $st + s - 1$ deletions.

Proof. Let $\boldsymbol{u}$ be a word of length $sn - st - s + 1$ and assume that there exist two words $\boldsymbol{c}'$ and $\boldsymbol{c}''$ from $C^{(s)}$ such that $\boldsymbol{u}$ can be obtained from either of them by deleting $st + s - 1$ symbols. There exist runs in $\boldsymbol{u}$ that are not a multiple of s. Denote by $\widetilde{\boldsymbol{u}}$ the word obtained from $\boldsymbol{u}$ by completing each run with symbols to the nearest length which is a multiple of s. Since every run in a codeword from $C^{(s)}$ is a multiple of s, $\widetilde{\boldsymbol{u}}$ is obtained from either $\boldsymbol{c}'$ and $\boldsymbol{c}''$ by deleting at most st symbols. Clearly, there exist two words in C that give rise to the same sequence of length $n - t$ after deletion of t symbols. This contradicts the fact that C is a t-deletion correcting code. $\square$

Remark 2.2. This theorem can be modified in order to obtain codes of lengths that are not a multiple of s. If we need a code of length $sn + r$ we just repeat the last symbol in any codeword $s + r$ times instead of s times.

Corollary 2.3. *Let C be a binary single deletion correcting code of length n. Then the code*

$$C^{(s)} = \{(\underbrace{c_1 \dots c_1}_{s}, \underbrace{c_2 \dots c_2}_{s}, \dots, \underbrace{c_n \dots c_n}_{s}) \mid (c_1, c_2, \dots, c_n) \in C\}$$

has length sn and corrects $2s-1$ deletions. There exist binary $(2s-1)$-deletions correcting codes of length $n = sm$ and cardinality $\geq 2^m/(m+1)$, for any $s \geq 1$.

These codes are poor for large values of n since they lie below the lower bound in Theorem 1.5. On the other hand, there exists an obvious decoding algortithm for $C^{(s)}$ that has the complexity of the decoding algorithm for C.

The codes defined in Theorem 2.1 and Corollary 2.3 are not maximal in the sense that there exist words from F^{sn} that can be added to $C^{(s)}$ without destroying the t-deletion correcting property. The number of runs in a word obtained from Theorem 2.1 does not exceed n while the code $C^{(s)}$ has length sn. This gives the possibilty of extending the code C by taking words with more than n runs. Let $\boldsymbol{x}$ and $\boldsymbol{y}$ be two words of length n having $r(\boldsymbol{x})$ and $r(\boldsymbol{y})$ runs, respectively. It is easily checked that if $r(\boldsymbol{x}) - r(\boldsymbol{y}) = 2d+1$ then $d_L(\boldsymbol{x}, \boldsymbol{y}) \geq 2d$. This follows by the fact that the deletion of a single symbol reduces the number of runs by at most 2.

In [10] a greedy search was performed in order to construct insertion/ deletion codes with Levenshtein distance $s > 2$. Starting from an arbitrary permutation of all 2^n words the authors pick up a word which is at Levenshtein distance at least 6 from all chosen words. This procedure is repeated for $5 \cdot 10^4$ starting permutations and the largest code is selected.

In the table that follows we present our results for 2-deletion correcting codes of length up to 14 compared with the codes obtained by Helberg-Fereira and Swart-Fereira. We used several different approaches to constructing such codes.

(A) Construction of a 2-deletion code as a subset of $VT_0(n)$, i.e. backtrack on the words of $VT_0(n)$.

(B) Given $C = VT_0(\frac{n}{2})$ we consider $C^{(2)}$ and try to add words to it by a greedy search on the words from $F^n \setminus C^{(2)}$.

(C) The same as in (B) but repeating the greedy search on 5.10^4 permutations of the words outside $C^{(2)}$.

(D) Greedy search performed on 5.10^4 random permutations of all words from F^n.

(E) Greedy search performed on various Gray codes.

(F) Exhaustive search (backtrack). For $n = 10, 11, 12$ the program has been terminated after 24 hours of computation. This is indicated by a question mark in the last column of the table below.

Surprisingly, the largest codes for $n = 13, 14$ have been produced by strategy (E).

n	[2]	[10]	(A)	(B)	(C)	(D)	(E)	(F)
4	2	2	2	2	2	2	2	2
5	2	2	2	2	2	2	2	2
6	3	4	4	2	2	4	4	4
7	4	5	4	4	4	5	5	5
8	5	7	5	5	5	7	6	7
9	6	10	7	8	9	10	10	11
10	8	14	12	10	12	14	14	(?)16
11	9	20	17	17	20	20	21	(?)20
12	11	29	26	25	28	29	31	(?)29
13	15	—	38	40	43	43	49	
14	18	—	59	63	63	65	72	

Below we list the 2-deletion correcting codes with parameters $(n, M) = (9, 11)$, $(10, 16)$, $(13, 47)$, $(14, 72)$ that were previously unknown.

The (9,11)-code.

000000000 111111111 000000111 000101100
000111111 001100010 011111000 101001001
110110110 111000000 111000111

The (10,16)-code.

0000000000 1111111111 0000000111 0000101010
0000111111 0001111000 0101010111 0110010001
0111110011 1000110011 1001010000 1100111101
1110000111 1110101010 1111000000 1111111000

The (11,21)-code.

01000011010 01000000011 11000000000 11000001111
11000110001 11001101110 11011010000 11011110011
11110001011 11111111000 11111111111 10101010101
10001111111 00001100111 00001111000 00010100100
00110100011 00111000000 00101110110 01110101111
01110110010

The (12,31)-code.

110111101010	110111111111	110111000111	110111000000
110101001100	111100111101	111101100001	111111111000
111001001011	111000100000	101000000011	101000011110
101001100010	100100111111	100111001001	100110101110
100011111000	100001001101	100000101000	000000000000
000000011111	000011000001	000011101010	000110110011
000111111110	000100010110	011000110111	011110000110
011100111000	010101010101	010111111001	

The (13,47)-code.

1000101110110	1000101010000	1000101000111	1000111110001
1000111111111	1000001011001	1000000111111	1000000001000
0000000000111	0000000110110	0000001110000	0000110000010
0000110011101	0000111111100	0000111010011	0001110010100
0011000010111	0011000110001	0011011011011	0011011110000
0011010000000	0011110000110	0011100111110	0010010101010
0110001001100	0110011100111	0110010000101	0110111111001
0111100111000	0111101001001	0111111011110	0101000111101
1100110110010	1101100101110	1101110100011	1101010111111
1101000110011	1111000000000	1111000000111	1111110000010
1111111101000	1111111111111	1111111001101	1111100011111
1110000101010	1010110000100	1011100110101	

The (14,72)-code.

11000100010000	11000100010111	11001100000011	11001100011010
11001100111111	11001101111000	11001111001011	11001010101001
11011001100000	11011110111101	11011111001100	11011100001111
11010010011100	11110000000001	11110001100111	11110001011000
11110110101010	11110111110111	11110101111100	11111100000010
11111101100011	11111111010000	11101000010101	11101011100001
11100010111101	11100001000110	10100000011001	10100000111110
10100011011011	10100011100100	10101100100101	10111111001111
10110101010111	10010010000010	10010110011110	10011101010000
10011111110110	10001001100011	10001110111001	10000101111111
10000010111000	10000000001111	10000000000001	00000000110100
00000001111011	00000001000110	00000110101001	00000111001110
00000111000000	00001100001011	00001101111010	00001111100011
00011001001000	00011010011111	00011111100000	00011111111111
00010101010110	00110000111100	00110011100010	00110110011001
00110100000001	00111100000110	00111110111000	00101111010101
00100001010111	01111010010001	01110100110110	01110011111110
01110011010011	01010000100000	01010101101000	01010100011101

The next table contains the cardinalities for the best t-deletion codes generated by us with $t = 2, 3, 4, 5$. The entries given with bold letters indicate the cardinalities of the optimal codes. The codes obtained by Helberg and Fereira [2] are given in the second, third and fourth column.

	Helberg-Fereira[2]			improved lower bounds			
n	$t=3$	$t=4$	$t=5$	$t=2$	$t=3$	$t=4$	$t=5$
4	2	—	—	**2**	**2**	—	—
5	2	2	—	**2**	**2**	**2**	—
6	2	2	2	**4**	**2**	**2**	**2**
7	2	2	2	**5**	**2**	**2**	**2**
8	3	2	2	**7**	**4**	**2**	**2**
9	4	2	2	**11**	**5**	**2**	**2**
10	4	3	2	16	6	**4**	**2**
11	5	4	2	21	7	5	**2**
12	6	4	3	31	10	5	**4**
13	8	4	4	49	12	5	5
14	8	5	4	75	15	7	5
15				109	24	9	5
16				176	31	12	7
17				286	48	15	7
18				485	71	21	9
19				813	103	26	12
20				1358	154	38	15
21				2299	242	49	18
22				3949	368	72	24
23				6787	579	100	32
24				11754	913	145	42
25				20491	1459	216	57
26				35858	2348	316	72
27				63035	3792	470	101
28				111176	6182	695	141
29				196932	10185	1057	193
30				350172	16776	1608	276

3. Upper bounds. In this section we prove several exact values and derive some upper bounds for the numbers $L_2(n,t)$.

Theorem 3.1. *$L_2(n,t) = 2$ for every $t = 1,2,\dots$ and every $n = t+1,\dots,2t+1$.*

This result is straightforward and does not require a proof.

Theorem 3.2. *$L_2(2t+2,t) = 4$ for every $t = 1,2,\dots$.*

P r o o f. Let C an t-deletion correcting code of length $2t+2$ and maximal cardinality. Denote by a_i the number the of words of (Hamming) weight i in C. Then $\sum_{j=0}^{t} a_j \le 1$ and $\sum_{j=t+2}^{2t+2} a_j \le 1$. Assume $a_{t+1} \ge 3$. Then there exist two words of weight $t+1$ having the same symbol in the first position, say 1. The Levenshtein distance between these words is obviously at most $2t$ since they share the common subsequence 10^{t+1}, a contradiction. Hence $a_{t+1} = 2$ and $L_2(2t+2,t) \le 4$.

The code

$$C = \{(\underbrace{0,0\dots,0}_{2t+2}),(\underbrace{0,\dots,0}_{t+1},\underbrace{1,\dots,1}_{t+1}),(\underbrace{1,\dots,1}_{t+1},\underbrace{0,\dots,0}_{t+1}),(\underbrace{1,1,\dots,1}_{2t+2})\}$$

is a t-deletion correcting code, which gives $L_2(2t+2,t) = 4$. $\square$

Theorem 3.3. *For every $t = 1,2,\dots,$ we have $5 \le L_2(2t+3,t) \le 6$.*

P r o o f. The code

$$C = \{(\underbrace{0,0\dots,0}_{2t+3}),(\underbrace{0,\dots,0}_{t+1},\underbrace{1,\dots,1}_{t+2}),(\underbrace{1,\dots,1}_{t+1},\underbrace{0,\dots,0}_{t+2}),(1,0,1,\dots,0,1),(\underbrace{1,1,\dots,1}_{2t+3})\}$$

is a t-deletion correcting code.

Assume $L_2(2t+3,t) \ge 7$ and let C be a binary t-deletion correcting code of length $2t+3$ and cardinality 7. Without loss of generality, we can assume that C contains the all-zero and the all-one words. The remaining five words are of weight $t+1$ and $t+2$.

Let us note first that C has at most one word of weight $t+1$ beginning with 1. If we assume that two such words exist then C is not a t-deletion correcting code since these words share the common subsequence 10^{t+2}. Similarly, there exist at most one word of each of the following types:

– weight $s+1$ and beginning with 00;

– weight $s+1$ and beginning with 01;

– weight $s+2$ and beginning with 0;

– weight $s+2$ and beginning with 10;

– weight $s+2$ and beginning with 11.

Without loss of generality we can assume that apart from 0^{2s+3} and 1^{2s+3}, C contains three words of weight $s+1$ and two words of weight $s+2$. Up to equvalence, we have three possibilities:

Case 1	*Case 2*	*Case 3*
$\boldsymbol{u}_1 = (1,*,*,\ldots,*,*)$	$\boldsymbol{u}_1 = (1,*,*,\ldots,*,*)$	$\boldsymbol{u}_1 = (1,*,*,\ldots,*,*)$
$\boldsymbol{u}_2 = (0,0,*,\ldots,*,*)$	$\boldsymbol{u}_2 = (0,0,*,\ldots,*,*)$	$\boldsymbol{u}_2 = (0,0,*,\ldots,*,*)$
$\boldsymbol{u}_3 = (0,1,*,\ldots,*,*)$	$\boldsymbol{u}_3 = (0,1,*,\ldots,*,*)$	$\boldsymbol{u}_3 = (0,1,*,\ldots,*,*)$
$\boldsymbol{u}_4 = (0,*,*,\ldots,*,*)$	$\boldsymbol{u}_4 = (1,0,*,\ldots,*,*)$	$\boldsymbol{u}_4 = (0,*,*,\ldots,*,*)$
$\boldsymbol{u}_5 = (1,1,*,\ldots,*,*)$	$\boldsymbol{u}_5 = (1,1,*,\ldots,*,*)$	$\boldsymbol{u}_5 = (1,0,*,\ldots,*,*)$

In all three cases, the words $\boldsymbol{u}_1$, $\boldsymbol{u}_2$, $\boldsymbol{u}_3$ are of weight $s+1$ and $\boldsymbol{u}_4$, $\boldsymbol{u}_5$ are of weight $s+2$.

We are going to rule out Case 1. First note that the second symbol of $\boldsymbol{u}_1$ is 0 (since otherwise $d_L(\boldsymbol{u}_1,\boldsymbol{u}_5) \le 2t$) and that the second symbol of $\boldsymbol{u}_4$ is 1 (since otherwise $d_L(\boldsymbol{u}_3,\boldsymbol{u}_4) \le 2t$). Moreover, the third symbols of $\boldsymbol{u}_3$ and $\boldsymbol{u}_4$ are different. Hence C contains the words:

$$\begin{aligned}
\boldsymbol{u}_1 &= (1,0,*,*,\ldots,*)\\
\boldsymbol{u}_2 &= (0,0,*,*,\ldots,*)\\
\boldsymbol{u}_3 &= (0,1,x,*,\ldots,*)\\
\boldsymbol{u}_4 &= (0,1,\overline{x},*,\ldots,*)\\
\boldsymbol{u}_5 &= (1,1,*,*,\ldots,*)
\end{aligned}$$

Now $\overline{x} \neq 0$ since otherwise $\boldsymbol{u}_2, \boldsymbol{u}_4$ contain the subsequence $0^2 1^{t+1}$ and $d_L(\boldsymbol{u}_1,\boldsymbol{u}_2) \le 2t$. Further the third symbol in $\boldsymbol{u}_1$ is 0 (compare $\boldsymbol{u}_1$ and $\boldsymbol{u}_5$). Hence the five words have the form

$$\begin{aligned}
\boldsymbol{u}_1 &= (1,0,0,x,\ldots,*)\\
\boldsymbol{u}_2 &= (0,0,y,*,\ldots,*)\\
\boldsymbol{u}_3 &= (0,1,0,z,\ldots,*)\\
\boldsymbol{u}_4 &= (0,1,1,*,\ldots,*)\\
\boldsymbol{u}_5 &= (1,1,*,*,\ldots,*)
\end{aligned}$$

Two of the symbols x, y, z are the same. This corresponding words are at Levenshtein distance $\leq 2t$, a contradiction. Case 2 and case 3 are ruled out in a similar way. $\square$

The exhaustive search performed for $n = 7$, $t = 2$ and $n = 9$, $t = 3$ gives $L_2(7, 2) = L_2(9, 3) = 5$ which suggests that we have generally $L_2(2t + 3, t) = 5$ for $t \geq 2$. In case of $t = 1$, we have Varshamov-Tenengolts codes that have length 5, cardinality 6 and correct 1 deletion.

Now we describe several upper bounds on $L_2(n, t)$. First we give a trivial recursive upper bound which is true for arbitrary alphabets.

Theorem 3.4. *$L_q(n + 1, t) \leq qL_q(n, t)$ for every $t = 1, 2, \ldots$. In particular, $L_2(n + 1, t) \leq 2L_2(n, t)$.*

The next upper bound was proved by Tolhuizen for any alphabet size q and any t [11]. It is a generalization of an earlier result by Levenshtein [7].

Theorem 3.5 [11]**.** *For any integer r with $1 \leq t \leq r + 1 \leq n$,*

$$L_q(n, t) \leq \frac{q^{n-t}}{\sum\limits_{i=0}^{t} \binom{r-t+1}{i}} + \frac{q \sum\limits_{i=0}^{r+2t-1} \binom{n+t-1}{i}(q-1)^i}{\sum\limits_{i=0}^{t} \binom{n+t}{i}(q-1)^i}.$$

While the bound from Theorem 3.5 is the best bound asymptically, the simple bound from Theorem 3.4 gives much better estimates for small n. For example, Tolhuizen's bound gives:

$$L_2(10, 2) \leq 58, \quad L_2(11, 2) \leq 100, \quad L_2(12, 2) \leq 172,$$

while by Theorem 3.4 in conjunction with the exact value $L_2(9, 2) = 11$, we get

$$L_2(10, 2) \leq 22, \quad L_2(11, 2) \leq 44, \quad L_2(12, 2) \leq 88.$$

Theorem 3.4 can be improved further. Asymptotically, the improvement is still worse than Tolhuizen's result, but for small lengths it yields estimates that are better than those obtained by Theorem 3.4 and Theorem 3.5.

Theorem 3.6. *For every $s = 0, 1, \ldots, \lfloor n/2 \rfloor$,*

$$L_2(n, t) \leq 2L_2(n - 2s - 1, t - s).$$

Proof. Let C be a t-deletion correcting code of length n and cardinality $L_2(n,t)$. The code C can be represented as $C = D_1 \cup D_2$ where D_1 is the set of codewords that have at most s 1's in the first $2s+1$ positions and D_2 is the set of codewords with at least $s+1$ 1's in the first $2s+1$ positions. The codes D_i must be $(t-s)$-deletion correcting codes. Hence

$$|C| = |D_1| + |D_2| \leq 2L_2(n-2s-1, t-s). \qquad \square$$

The table below compares the upper bounds for t-deletion correcting codes with $t = 2, 3, 4, 5$ and $10 \leq n \leq 20$ obtained from Theorem 3.5 on one side and Theorems 3.3 and 3.6 on the other.

n	$t=2$		$t=3$		$t=4$		$t=5$	
	T 3.5	T 3.6	T 3.5	T 3.6	T 3.5	T 3.6	T 3.5	T 3.6
10	58	22	23	10	—	—	—	—
11	100	44	37	14	18	6	—	—
12	172	88	60	22	27	10	—	—
13	301	176	99	44	42	14	21	6
14	530	352	165	88	67	22	32	10
15	940	704	279	176	107	44	49	14
16	1678	1408	473	352	174	88	77	22
17	3015	2816	811	704	285	176	120	44
18	5447	5632	1399	1408	471	352	191	88
19	9890	11264	2431	2816	786	704	307	176
20	18037	22528	4253	5632	1321	1408	497	352

Acknowledgements. The research of the first author has been supported by the Bulgarian NSRF under Contract I-1304/03.

REFERENCES

[1] Ginzburg B. D. A number-theoretic function with an application in the theory of coding. *Problemy Kibernetiki* **19** (1967), 249–252 (in Russian); English translation: *Systems Theory Research* **19** (1970), 255–259.

[2] Helberg A. S. J., H. C. Fereira. On multiple insertion/deletion correcting codes. *IEEE Trans. Inf. Theory* **48** (2002), 305–308.

[3] Hirschberg D. S., M. Reigner. Tight bounds on the number of string subsequences. *J. of Disc. Algorithms* **1** (2000), 123–132.

[4] Levenshtein V. Binary codes capable of correcting deletions, insertions and reversals. *Dokl. Akad. Nauk SSSR* **163** (1965), 845–848 (in Russian); English translation: *Soviet Phys.-Dokl.* **10** (1966), 707–710.

[5] Levenshtein V. On perfect codes in the insertion/deletion metric. *Diskr. Mat.* **3** (1991), 3–20. (in Russian); English translation: *Discrete Math. and Applications* **2** (1992), 241–258.

[6] Levenshtein V. Efficient reconstruction of sequences from their subsequences or supersequences. *J. Comb. Theory Ser. A* **93** (2001), 310–332.

[7] Levenshtein V. Bounds for insertion-deletion-correcting codes. IEEE Int Symp. on Inf. Theory 2002, Lausanne, Switzerland.

[8] Martirosyan S. Single-error correcting close packed and perfect codes. In: Proc. of the 1st INTAS Int. Seminar on Coding Theory and Combinatorics, Thakhadzor, Armenia, 1996, 90–115.

[9] Sloane N. J. A. On single-deletion-correcting codes. In: Codes and Designs, Ohio State University, May 2000 (Ray-Chaudhuri Festschrift), (eds K. T. Arasu, A. Seress), Walter de Gruyter, Berlin, 2002, 273–291.

[10] Swart T. G., H. C. Fereira. A note on double insertion/deletion correcting codes. *IEEE Trans. Inf Theory* **49** (2003), 269–272.

[11] Tolhuizen L. Upper bounds on the size of insertion/deletion-correcting codes. Proc. of the 8th Workshop on ACCT, Tsarskoe Selo, Russia, 2002, 242–246.

[12] Varshamov R. R. On an arithmetic function with an application in the theory of coding. *Dokl. Akad. Nauk SSSR* **161** (1965), 540–543 (in Russian).

[13] Varshamov R. R., G. M. Tenengolts. Codes which correct a single asymmetric error. *Avtomatika i Telemehanika* **26**, No 2, (1965), 288–292 (in Russian); English translation: *Automation and remote control* **26** (1965), 286–290.

Ivan Landjev
New Bulgarian University
21 Montevideo str.
1618 Sofia, Bulgaria
e-mail: `ilandjev@nbu.bg`
and
Institute of Mathematics and Informatics
Bulgarian Academy of Sciences
Acad. G. Bonchev str., Bl. 8
1113, Sofia, Bulgaria
e-mail: `ivan@moi.math.bas.bg`

Kristiyan Haralambiev
Department of Computer Science
Courant Institute of Mathematical Sciences
New York University, USA

Received February 20, 2006
Final Accepted February 23, 2007