A Study of Error Correcting Code using Impartial Games

Author: Yifu Zhang[1]

Mentor: Prof. Noah Aydin[2]

Abstract

Lexicographic codes (abbreviated as lexicode) are defined as greedily generated codes with fixed minimal distance. Surprisingly, despite simple construction, such codes often have strong bounds. Many of the known optimal codes are lexicographic, such as the well-known binary Hamming codes, binary Golay code, etc. This paper will study lexicographic codes by looking at the Sprague-Grundy analysis of the game Mock Turtle, which generates the extended Hamming code as well as its shortened version. This paper aims to shed light on the generation of lexicode with specified minimal distance d by first looking at the special case of Mock Turtle and extended Hamming code, which can then be generalized to find other families of lexicographic code.

Keywords: error-correcting code, lexicographic code, impartial game

2021 S.-T. Yau High School Science Award

[1]From Keystone Academy, Beijing, China
[2]From Kenyon College, Ohio, USA

Contents

2021 S.-T. Yau High School Science Award

1 Introduction

One of the pioneering research on the subject of lexicographic code is by Conway and Sloane[1], who analyzed this subject using Sprague-Grundy theory of impartial games, most notably the Grundy's game . Grundy's game is defined as an impartial heap game under normal play rules, where a legal move is to split any heap into strictly different sizes. To study this game, we will first review the theory of impartial games. One of the defining texts of the subject is [3] by Berlekamp, Conway and Guy, while in this paper the author found [2] very helpful and studied combinatorial game theory using this text. The Sprague-Grundy theory analyzes impartial games by assigning G-values to each game positions, which is stated below[1] [2] :

1. The zero game has G-value 0.
2. $G(P) = \text{mex}\{G(Q), G(R)...\}$, where $Q, R, ...$ are all the game positions that can be obtained from P using a single move. [3]
3. A player win by consistently move to positions with G-value 0.
4. The G-value of a general position $P_1 + P_2 + P_3...$ is $\sum_{i=1} G(P_i)$, where addition is bit-wise addition(also known as nim addition or exclusive-or), which will be denoted as $\oplus$

By analyzing Grundy's game using the above S-G theory, Conway and Sloane deduced that the codewords in lexicodes corresponds to winning positions in Grundy's game. However, a complete S-G analysis of Grundy's game has been an open problem, In particular, the periodicity of G-value of the atomic positions remains to be proven, which hinders the development of lexicodes

Later, Brualdi and Pless [4] provided a different approach to generate lexicographic code by assigning g-value (similar but distinct from the previous G-value) to each vector of length n over $GF(2)$, and the lexicographic code would constitute of those with G-value 0. Trachtenberg, on the other hand, developed an iterative algorithms, namely the $\mathcal{B}$-construction, that constructs generator matrices of lexicographic codes with desired minimal distance and also have low trellis decoding complexity, which made it helpful for very large size integration based decoding.[8] He also suggested that the lexicodes can be

[3]mex$\{S\}$ is the minimal excludant operation, which finds the minimum non-negative integer that is not in S. e.g. $\text{mex}\{1, 2, 3\} = 0$

2021 S.-T. Yau High School Science Award

view as heuristically "approximation" of optimal codes, because they are "usually within one of the optimal minimal distance" [8]. The linearity of lexicodes is also widely known due to works of several different authors, including [1] and [4].

The game Mock Turtle, which turns out to be highly relevant in the study of lexicode, is first discussed in [3], where a complete Nim analysis is presented. The game Mock Turtle is a coin turning game under normal play. A position in the game is denoted with a binary string, where 1 corresponds to head and 0 correspond tail. A move in the game is to flip a coin from head to tail, and then flip up to 2 coins (or no coins) on the right of the flipped coin either from head to tail, or tail to head. The lexicodes, which we will shortly discuss about, is the Extended Hamming codes.

It should also be noted that lexicodes are not restricted to be binary. In fact, it has been proven that for base 2^a, unrestricted lexicodes are closed under bit-wise addition, and for base 2^{3^a}, unrestricted lexicodes are closed under nim-multiplication[1]. However, for the interest of this paper, we will restrict our analysis to binary lexicodes.

In this paper, we aim to generalize the game Mock Turtle to any minimal distance d, which will then be used to generate the parity-check matrix of lexicographic code over $GF(2)$ of any minimal distance, provided the analysis of Mock Turtle with such minimal distance is complete.

2 Generalization of Mock Turtle

In this section, we will analyze Mock Turtle using Sprague-Grundy theory. We begin by generalizing the definition of Mock Turtle:

Definition 2.1. *Mock Turtle for distance $d > 2$ and length n is a n coin turning game, with head denoted as 1 and tail as 0. A legal move is to flip one coin from head to tail and flipping less than $d-1$ coins to the right of that coin. In other words, he may flip at most $d-1$ coins provided the leftmost flip is from head to tail. The last player who has no legal moves loses. To denote such a game, we use a binary vector $a_n a_{n-1}...a_0$, where a_n = 1 or 0 denotes the position of nth coin. The unit vectors(atomic positions) are denoted $\hat{\imath}_k$, and $\hat{\imath}_0 = 00...01$.* [4]

[4]Due to the nature of this subject, we will use binary vectors and position interchangeably throughout the paper

2021 S.-T. Yau High School Science Award

For example, at position 011000 for Mock Turtle of distance 4, the player can move to 001000 by only turning one coin from head(1) to tail(0), or move to 001100 by turning the leftmost head to tail, and then flip the 3rd coin counting from right from tail to a head,

Mock Turtle of minimal distance 4 and 3(called Turning Turtle) has been analyzed thoroughly by Berlekamp, Conway and Guy. Their results is presented as the following theorem:

Theorem 2.1. *[3, p.463] For Mock Turtle of minimal distance 3, $G(\hat{\imath}_k) = k+1$. For Mock Turtle of minimal distance 4, $G(\hat{\imath}_k) = k'$, where k' is obtained by expressing k in binary, then add an overall parity check digit on the right so that the Hamming weight[5] of the string is odd.*

proof. We will only look at the $d = 3$ (Turning Turtle) case for simplicity. For the $d = 4$ case, see [3, p.464].

In the Turning Turtle case: by strong induction: base case: $G(\hat{\imath}_0) = 1$, $G(\hat{\imath}_1) = 2$. Suppose for all $0 \leq j \leq k$, $G(\hat{\imath}_j) = j + 1$. For $\hat{\imath}_{k+1}$, it can reach any $\hat{\imath}_j$ within one move. Any general position $P = P_k P_{k-1} ... P_1$, where P_j is either 0 or 1 has G-value $P_0 G(\hat{\imath}_0) \oplus ... \oplus P_k G(\hat{\imath}_j) = 1 \times P_0 \oplus ... \oplus P_k(k+1) \leq k+1$ by definition of bit-wise addition. Hence, $G(\hat{\imath}_{k+1}) = \operatorname{mex} 0, 1, ...k+1 = k+2$

$\mathbb{Q}.\mathbb{E}.\mathbb{D}$

The following lemma is required to generate G-value for Mock Turtle of minimal distance d.

Lemma 2.1. *For any position P in Mock Turtle of minimal distance d, it can reach another position Q within one legal move if and only if:*

1. $\mathrm{wt}(P + Q) \leq d - 1$[6] *and*

2. *The binary number P is greater than the binary number Q.*

proof. If Q can be obtained in a legal move from P, then $P > Q$ because the leftmost coin must be flipped from 1 to 0. In addition, $\mathrm{wt}(P + Q) \leq d - 1$ because only $d - 1$ coins can be flipped.

If for some Q and P there is $\mathrm{wt}(P + Q) \leq d - 1$, and $P > Q$, then Q can be obtained in a legal move from P. This follows directly from the definition of Mock Turtle.

[5]Hamming weight of a binary string is the number of 1s in the string. e.g 0011 has weight 2.

[6]$\mathrm{wt}(P + Q)$ is the Hamming distance between P and Q, i.e. how many digits do P and Q differs.

2021 S.-T. Yau High School Science Award

Q.E.D

Lemma 2.2. *For Mock Turtle of distance d, $G(\hat{\imath}_k) = 2^k$ for $k < d-1$.*

proof. Let $\hat{\imath}_k$ be an arbitrary unit vector with $k < d-1$. By lemma 2.1, all the positions less than 2^k can be reached from $\hat{\imath}_k$ because the maximum weight of such positions is $d-2$, so the maximum Hamming distance between such positions and $\hat{\imath}_k$ is $d-1$. These positions have G-value ranging from 0 to $2^k - 1$, hence $G(\hat{\imath}_k) = 2^k$

Q.E.D

Algorithm 2.1. *The algorithm that generates G-value of unit vectors in Mock turtle of length n and distance d follows:*

1. *$G(\hat{\imath}_k) = 2^k$ for $k < d-1$.*
2. *For all general position $P = P_0P_1P_2...P_k$ of length $k+1$ such that $\mathrm{wt}(P) < d-1$, compute $G(P) = \sum_{i=1} G(\hat{\imath}_i)P_i$. We will call the set of all computed G-values $\mathcal{G}$.*
3. *$G(\hat{\imath}_{k+1}) = \mathrm{mex}\{\mathcal{G}\}$. Include $G(\hat{\imath}_{k+1})$ in $\mathcal{G}$.*
4. *Repeat step 2 and 3 to generate all unit vectors up to $\hat{\imath}_{n-1}$.*

proof. $G(\hat{\imath}_k) = k+1$ for $k < 2$ by lemma 2.2.

By definition of G-value, $G(\hat{\imath}_k) = \mathrm{mex}\{\mathcal{G}\}$, where $\mathcal{G}$ is the set of G-value for all positions obtainable from $\hat{\imath}_k$. Hence we only need to prove that $\mathcal{G}$ is indeed such set.

Define $\mathcal{P}$ as the set of all positions that are obtainable in one move. Since P has a maximum length $k+1$, $P < \hat{\imath}_{k+1}$. In addition, $\mathrm{wt}(P) < d-1$, so $\mathrm{wt}(P + \hat{\imath}_{k+1}) \leq d-1$. Thus, by lemma 2.1, the set of all P is $\mathcal{P}$. By definition of G-value, $\mathcal{G}$ is the set of G-values for all elements in $\mathcal{P}$.

Q.E.D

We present the Python 3.9 implementation of this algorithm in Appendix A. In essence, this iterative algorithm will first find G-value of all the positions that can be obtained in one move from a unit vector using previously generated G-values, then compute G-value of the next unit vector by finding the minimal excludant.

2021 S.-T. Yau High School Science Award

3 Construction of Lexicodes Using Mock Turtle

In this section, we aim to construct the parity check matrix of n, k, d lexicode with predetermined n, d.

3.1 Parity Check Matrix of Lexicode

We first define g-value, which should be carefully distinguished from G-value in game theory, It is first introduced by Brualdi and Pless[4]

Definition 3.1. *Let d be an integer with $0 \leq d \leq n$. Assume that the vectors in F_2^n have been listed in some order(in our case, lexicographic order): $z_1, z_2, ...z_{2^n}$. We recursively define*

$$g : F_2^n \to \mathbb{Z}^+ \cup 0$$

as follows.

1. $g(z_1) = 0$.
2. *$g(z_i)$ is the smallest non-negative integer t such that $wt(z_i + x) \geq d$ for all vector x in $\{z_i, ...z_{i-1}\}$ which satisfy $g(x) = 4$. If no such t exists we define $g(z)$ to be the smallest integer not in $\{g(z_1), .., , g(z_{i-1})\}$.*

We reword this definition by utilizing the minimal excludant operation.

Definition 3.2. *Let d be an integer with $0 \leq d \leq n$. Assume that the vectors in F_2^n have been listed in some order(in our case, lexicographic order): $z_1, z_2, ...z_{2^n}$. We recursively define*

$$g : F_2^n \to \mathbb{Z}^+ \cup 0$$

as follows.

1. $g(z_1) = 0$.
2. *$g(z_i) = \text{mex}\{G\}$, where G is the set of g-values of all $x \in \{z_1, ...z_{i-1}\}$ such that $\text{wt}(z_i + x) < d$.* [7] .

Theorem 3.1. *[4] Let $\mathcal{B}$ be an ordered basis of F_2^n and let d be an integer with $3 \leq d \leq n$[8]. Then a parity check matrix for n for the $\mathbb{B}$-greedy code C is*

$$H = \begin{bmatrix} g(\hat{\imath}_n) & ... & g(\hat{\imath}_2) & g_(\hat{\imath}_1) \end{bmatrix}$$

[7]In other word, $g(z_i)$ is the minimal excludant of set of all g-values of its previous vectors x that has a distance less than d with z_i

[8]The original theorem states $0 \leq d \leq n$, but for the purpose of this paper we restrict it to 3

2021 S.-T. Yau High School Science Award

where g-values are in binary.

proof. The proof is omitted and can be found in [4]

$\mathbb{Q.E.D}$

Before we present one of the main theorems in this paper, one lemma is needed:

Lemma 3.1. *Let $z \in F_2^n$ and let d be an integer with $3 \leq d \leq n$, then $G(z) = g(z)$, where $G(z)$ is the Mock Turtle of length n and distance d.*

proof. Assume F_2^n is listed in lexicographic order. Let z_k be an arbitrary vector in F_2^n and $\mathcal{P}$ be the set of all positions that is obtainable in one move from z, G as defined in definition 3.2, and $\mathcal{G}$ be the set of G-values of all elements in $\mathcal{P}$. Thus, we need to prove $G = \mathcal{G}$. By induction: In lexicographic order, $g(z_1) = 0 = G(0)$, $g(z_2) = 1 = G(1)$. Suppose for some $0 \leq j \leq k$, $g(z_k) = G(z_k)$. To prove the case for $G(z_{k+1})$, let $x \in z_1, ..., z_{i-1}$ be any vector satisfies $\text{wt}(z_{k+1} + x) \ < d$, and in lexicographic order $z_{k+1} < x$ for all x, by lemma 2.1 the set of all x is $\mathcal{P}$, thus $G = \mathcal{G}$.

$\mathbb{Q.E.D}$

By substituting $g(\hat{\imath}_i)$ with $G(\hat{\imath}_i)$ in theorem 3.1, we derive the following lemma:

Lemma 3.2. *Let F_2^n be lexicographically ordered. The parity check matrix for $[n, k, d]$ lexicode C is*

$$H = \begin{bmatrix} G(\hat{\imath}_n) & ... & G(\hat{\imath}_2) & G_(\hat{\imath}_1) \end{bmatrix}$$

where G-values are in binary.

This result directly generates n, k, d lexicode.

Algorithm 3.1. *The algorithm that generates the parity check matrix H of $[n, k, d]$ lexicode follows:*

1. *Apply algorithm 2.1 with length n and distance d*
2. *$H = \begin{bmatrix} G(\hat{\imath}_n) & ... & G(\hat{\imath}_2) & G_(\hat{\imath}_1) \end{bmatrix}$, where G-values are in binary*

proof. Step 1 generates the G-values of unit vectors for Mock Turtle of length n and distance d, and Step 2 follows directly from lemma 3.2.

2021 S.-T. Yau High School Science Award

Q.E.D

The parameter of the lexicode generated in this manner can be found by the theorem below:

Theorem 3.2. *The parameter of lexicode generated by algorithm 3.1 is* $[n, n-m, d]$, *where* m *is length of* $G(\hat{\imath}_n)$ *in binary.*

proof. By lemma 3.2, the first column of a parity check matrix for the code is $G(\hat{\imath}_n)$. Since parity check matrix is $(n-k) \times n$, $m = n-k$, and $k = n-m$.

Q.E.D

The author implemented these two results as a script in Python 3.9 in Appendix A.

We will present this result using the example of Extended Hamming code.

Theorem 3.3. $[n, k, 4]$ *lexicode is optimal. Specifically, for any* $[n, k, 4]$ *lexicode with* $n > 5$, *no* $[n, k, 5]$ *linear code exist by sphere packing bound i.e.*

$$1 + n + \frac{n(n-1)}{2} > 2^{n-k}$$

. *In addition, for any* $[n, k, 4]$ *lexicode with* $n > 5$, *no* $[n, k+1, 4]$ *code exists by sphere packing bound, i.e.*

$$1 + n > 2^{n-k-1}$$

proof. By contradiction: Suppose for some $n > 5$, there is

$$1 + n + \frac{n(n-1)}{2} \leq 2^{n-k}$$

.

Then by theorem 3.2,

$$1 + n + \frac{n(n-1)}{2} \leq 2^{m+2} = 4 \cdot 2^m$$

where $2^{m+1} > n-1 \geq 2^m$. Thus

$$1 + n + \frac{n(n-1)}{2} \leq 4(n-1)$$

Simplification gives

$$n^2 - 7n + 10 \leq 0$$

2021 S.T. Yau High School Science Award

and hence $2 \leq n \leq 5$. This contradicts the hypothesis of $n > 5$. Thus any $[n, k, 5]$ does not satisfy the sphere packing bound.

By contradiction: suppose for some n, $[n, k+1, 4]$ lexicode exists. By theorem 3.2, the parameters of a $[n, k+1, 4]$ lexicode is $[n, n - \lfloor \log_2(n) \rfloor + 2, 4]$. By sphere packing bound:

$$1 + n < 2^{n-n+\lfloor \log_2(n) \rfloor - 2}$$

Simplifying gives:

$$\log_2(1+n) < \lfloor \log_2(n) \rfloor$$

which is impossible since $\log_2(1+n) > \log_2(n) \geq \lfloor \log_2(n) \rfloor$

Therefore, no $[n, k+1, 4]$ code exists.

$\mathbb{Q}.\mathbb{E}.\mathbb{D}$

This shows that the $[n, k, 4]$ lexicode is indeed optimal, and it is in fact the extended Hamming Code and its shortened version.

3.2 Generalization to minimal distances 5 and 6

Although the $[n, k, d]$ lexicode can be determined using algorithm 3.1, the time complexity of the algorithm hinders further studies. Nevertheless, the case of $[n, k, 5]$ and its extended code $[n, k, 6]$ is computed and compared with the known optimal bounds. 55 of the $[n, k, 5]$ and $[n, k, 6]$ new lexicodes matches the optimal bounds provided in [6] and are optimal linear codes. Among the non-optimal codes, 222 of them have dimension 1 less than the known bounds. All computed codes are within 3 of the known optimal dimension.A detailed list of such codes can be found in Appendix B.

Remark 3.1. *The $[n, k, 5]$ lexicodes have particularly strong dimension k for smaller n. For example, for $n \leq 100$, all codes are either optimal or 1 less than the optimal bound, among which 49 are optimal.*

Remark 3.2. *As pointed out in [1], the $[18, 9, 5]$ lexicode generated in this manner is the binary quadratic residue code, and $[18, 9, 6]$ is its extended code. Thus we have provided a new way to generate the binary quadratic residue code.*

Remark 3.3. *The computation of k in lexicodes of minimal distance 5 provides a empirical justification for lexicode being "heuristic approximation" of optimal codes, which was stated in [8].*

2021 S.-T. Yau High School Science Award

4 Conclusion and Future Work

This paper studies binary lexicographic code of general minimal distance by studying its related game Mock Turtle via Sprague-Grundy analysis. We generated the parity check matrix of binary lexicographic codes by using g-value of Mock Turtle. Some of our future work direction includes:

1. We would like to reduce the time complexity of Algorithm 3.1, which largely hinders the study of this topic. To do so one research direction is to solve the game Mock Turtle for a general minimal distance.

2. We would like to prove/disprove that there exists infinitely many $[n, k, 5]$ optimal lexicode. In addition, Are there infinitely many optimal lexicodes of any minimal distance?

5 Acknowledgment

The author would like to express his sincerest gratitude to Professor Noah Aydin for providing his generous support, extensive knowledge and valuable advice for my research.

The research was mainly conducted under the capacity of Pioneer Research Program, a program aims to mentor high school students to experience independent academic research. Prof. Noah Aydin was the author's mentor, who taught coding theory background knowledge to the author. The author then chose lexicographic code as his specific research topic due to his interest in combinatorial game theory, and then researched and read the background materials with the help from Prof. Aydin. The author created the algorithm for Mock Turtle, proved and computes the all results presented in the paper unless cited. After the program, the author decided to participate in the Yau award, and Prof. Aydin generously offered his mentorship free of charge.

The mentor of this paper, Prof. Noah Aydin is professor of mathematics at Kenyon College, Ohio, and has his primary research interest in Algebraic coding theory. His generous and enormous support is the most appreciated by the author.

2021 S.-T. Yau High School Science Award

References

[1] J. Conway and N. Sloane, “Lexicographic codes: Error-correcting codes from game theory,” in IEEE Transactions on Information Theory, vol. 32, no. 3, pp. 337-348, May 1986, doi: 10.1109/TIT.1986.1057187.

[2] M. H. Albert, R. J. Nowakowski, D. Wolfe, ”Lessons in Play: an Introduction to Combinatorial Game Theory”, by A K Peters, 2007, ISBN: 978-1-4398-6437-1

[3] E. R. Berlekamp, J. H. Conway, R. K. Guy, ”Winning Ways for Your Mathematical Plays”, 3 vols, by Massachusetts, A. K. Peters, 2001, ISBN: 1-56881-130-6

[4] R. A. Brualdi and V. Pless, ”Greedy Codes,” Proceedings. IEEE International Symposium on Information Theory, San Antonio, TX, USA, 1993, pp. 366-366, doi: 10.1109/ISIT.1993.748682.

[5] OEIS Wiki, by OEIS Foundation, https://oeis.org/wiki/Orderings#: :text= When%20applied%20to%20numbers%2C%20lexicographic,ordered%20by %20their%20smallest%20elements.

[6] “Generate Parameter Table for Linear Codes”, MinT, University of Salzberg, http://mint.sbg.ac.at/table.php?i=c&var=q-T-%CE%BB-t-d-m-n-k&miny=1&minx=1&dx=30&dy=17&mode=bd&opt=&b=2&de=1&p=sdn&col=1

[7] V. Pless, ”Introduction to the Theory of Error-Correcting Codes, 3rd ed, by Wiley Interscience, 1998

[8] A. Trachtenberg, ”Designing lexicographic codes with a given trellis complexity,” in IEEE Transactions on Information Theory, vol. 48, no. 1, pp. 89-100, Jan. 2002, doi: 10.1109/18.971740.

2021 S.-T. Yau High School Science Award

Appendices

A Code for G-value Generation

The following python codes generates all $G(\hat{\imath}_k)$ for Mock Turtle of length n and minimal distance d.

```

import itertools

def minimal_excludant(lst):
#finds minimal excludant of a tuple
  lst = set(lst)
  mex = 0
  while mex in lst:
    mex += 1
  return mex


def nim_sum(tup):
#computes nim-sum of all elements in tuple
  nimber =0
  for g in tup:
    nimber = nimber ^ g
  return nimber

def g_generator(d, lst):
#computes all G(P), where P can be obtained
#within one move from the next unit vector
  comb = []
  sum_all = []
  for i in range(2,d-1):
    comb.extend(list(itertools.combinations(lst, i)))
  sum_all.extend((nim_sum(tup) for tup in comb))
return sum_all

def g_value(n, d):
#Output a list the contains g-value for unit vectors
  G = [0]
  G.extend(2 ** i for i in range(0,d-1))
  ATOMIC_G = [2 ** i for i in range(0,d-1)]
  for i in range(d-1, n):
    G.extend(g_generator(d, ATOMIC_G))
    mex = minimal_excludant(G)
    ATOMIC_G.append(mex)
```

ward
2021

```
    G.append(mex)
  print(ATOMIC_G)
  return ATOMIC_G
```

We present a sample here:

```
  #Input: 8,4
  #Output: [1, 2, 4, 7, 8, 11, 13, 14]
```

The nth element in the list is $G(\hat{\imath}_n)$, e.g $\hat{\imath}_0 = 1$.

The python implementation to generate the dimension of lexicode given n, d follows:

```
def dimension_generator(n,d):
  k = n - len(bin(g_value(n,d)[-1]))+2
  print(k)
  return k
```

Sample:

```
  #Input: 8,4
  #Output: 4
```

B Dimension of $n, k, 5$ Lexicode

The optimal $n, k, 5$ lexicodes from $n = 4$ to $n = 512$, a total of 55 codes, are listed below:

[4, 0, 5]	[5, 1, 5]	[6, 1, 5]	[7, 1, 5]	[8, 2, 5]
[9, 2, 5]	[10, 3, 5]	[11, 4, 5]	[12, 4, 5]	[13, 5, 5]
[14, 6, 5]	[15, 7, 5]	[16, 8, 5]	[17, 9, 5]	[18, 9, 5]
[19, 10, 5]	[20, 11, 5]	[21, 12, 5]	[24, 14, 5]	[25, 15, 5]
[26, 16, 5]	[27, 17, 5]	[28, 18, 5]	[29, 19, 5]	[34, 23, 5]
[35, 24, 5]	[36, 25, 5]	[37, 26, 5]	[38, 27, 5]	[48, 36, 5]
[49, 37, 5]	[50, 38, 5]	[51, 39, 5]	[52, 40, 5]	[66, 53, 5]
[67, 54, 5]	[68, 55, 5]	[69, 56, 5]	[82, 68, 5]	[83, 69, 5]
[84, 70, 5]	[85, 71, 5]	[86, 72, 5]	[87, 73, 5]	[88, 74, 5]
[89, 75, 5]	[90, 76, 5]	[91, 77, 5]	[92, 78, 5]	[152, 136, 5]
[153, 137, 5]	[154, 138, 5]	[155, 139, 5]	[156, 140, 5]	[266, 248, 5]

and their extended codes.

The codes with dimension one less than the optimal bounds, are listed below:

2021 S.-T. Yau High School Science Award

[22, 12, 5]	[23, 13, 5]	[30, 19, 5]	[31, 20, 5]	[32, 21, 5]	[33, 22, 5]
[39, 27, 5]	[40, 28, 5]	[41, 29, 5]	[42, 30, 5]	[43, 31, 5]	[44, 32, 5]
[45, 33, 5]	[46, 34, 5]	[47, 35, 5]	[53, 40, 5]	[54, 41, 5]	[55, 42, 5]
[56, 43, 5]	[57, 44, 5]	[58, 45, 5]	[59, 46, 5]	[60, 47, 5]	[61, 48, 5]
[62, 49, 5]	[63, 50, 5]	[64, 51, 5]	[65, 52, 5]	[70, 56, 5]	[71, 57, 5]
[72, 58, 5]	[73, 59, 5]	[74, 60, 5]	[75, 61, 5]	[76, 62, 5]	[77, 63, 5]
[78, 64, 5]	[79, 65, 5]	[80, 66, 5]	[81, 67, 5]	[93, 78, 5]	[94, 79, 5]
[95, 80, 5]	[96, 81, 5]	[97, 82, 5]	[98, 83, 5]	[99, 84, 5]	[100, 85, 5]
[101, 86, 5]	[102, 87, 5]	[103, 88, 5]	[104, 89, 5]	[105, 90, 5]	[106, 91, 5]
[107, 92, 5]	[108, 93, 5]	[109, 94, 5]	[110, 95, 5]	[111, 96, 5]	[112, 97, 5]
[113, 98, 5]	[114, 99, 5]	[115, 100, 5]	[116, 101, 5]	[117, 102, 5]	[118, 103, 5]
[119, 104, 5]	[120, 105, 5]	[129, 113, 5]	[130, 114, 5]	[131, 115, 5]	[132, 116, 5]
[133, 117, 5]	[134, 118, 5]	[135, 119, 5]	[136, 120, 5]	[137, 121, 5]	[138, 122, 5]
[139, 123, 5]	[140, 124, 5]	[141, 125, 5]	[142, 126, 5]	[143, 127, 5]	[144, 128, 5]
[145, 129, 5]	[146, 130, 5]	[147, 131, 5]	[148, 132, 5]	[149, 133, 5]	[150, 134, 5]
[151, 135, 5]	[157, 140, 5]	[158, 141, 5]	[159, 142, 5]	[160, 143, 5]	[161, 144, 5]
[162, 145, 5]	[163, 146, 5]	[164, 147, 5]	[165, 148, 5]	[166, 149, 5]	[167, 150, 5]
[168, 151, 5]	[169, 152, 5]	[170, 153, 5]	[171, 154, 5]	[172, 155, 5]	[173, 156, 5]
[174, 157, 5]	[175, 158, 5]	[176, 159, 5]	[177, 160, 5]	[178, 161, 5]	[179, 162, 5]
[180, 163, 5]	[181, 164, 5]	[182, 165, 5]	[183, 166, 5]	[184, 167, 5]	[185, 168, 5]
[186, 169, 5]	[187, 170, 5]	[188, 171, 5]	[189, 172, 5]	[190, 173, 5]	[191, 174, 5]
[192, 175, 5]	[193, 176, 5]	[194, 177, 5]	[195, 178, 5]	[196, 179, 5]	[197, 180, 5]
[198, 181, 5]	[199, 182, 5]	[200, 183, 5]	[201, 184, 5]	[202, 185, 5]	[203, 186, 5]
[258, 240, 5]	[259, 241, 5]	[260, 242, 5]	[261, 243, 5]	[262, 244, 5]	[263, 245, 5]
[264, 246, 5]	[265, 247, 5]	[267, 248, 5]	[268, 249, 5]	[269, 250, 5]	[270, 251, 5]
[271, 252, 5]	[272, 253, 5]	[273, 254, 5]	[274, 255, 5]	[275, 256, 5]	[276, 257, 5]
[277, 258, 5]	[278, 259, 5]	[279, 260, 5]	[280, 261, 5]	[281, 262, 5]	[282, 263, 5]
[283, 264, 5]	[284, 265, 5]	[285, 266, 5]	[286, 267, 5]	[287, 268, 5]	[288, 269, 5]
[289, 270, 5]	[290, 271, 5]	[291, 272, 5]	[292, 273, 5]	[293, 274, 5]	[294, 275, 5]
[295, 276, 5]	[296, 277, 5]	[297, 278, 5]	[298, 279, 5]	[299, 280, 5]	[300, 281, 5]
[301, 282, 5]	[302, 283, 5]	[303, 284, 5]	[304, 285, 5]	[305, 286, 5]	[306, 287, 5]
[307, 288, 5]	[308, 289, 5]	[309, 290, 5]	[310, 291, 5]	[311, 292, 5]	[312, 293, 5]
[313, 294, 5]	[314, 295, 5]	[315, 296, 5]	[316, 297, 5]	[317, 298, 5]	[318, 299, 5]
[319, 300, 5]	[320, 301, 5]	[321, 302, 5]	[322, 303, 5]	[323, 304, 5]	[324, 305, 5]
[325, 306, 5]	[326, 307, 5]	[327, 308, 5]	[328, 309, 5]	[329, 310, 5]	[330, 311, 5]
[331, 312, 5]	[332, 313, 5]	[333, 314, 5]	[334, 315, 5]	[335, 316, 5]	[336, 317, 5]
[337, 318, 5]	[338, 319, 5]	[339, 320, 5]	[340, 321, 5]	[341, 322, 5]	[342, 323, 5]

2021 S.-T. Yau High School Science Award

and their extended codes

2021 S.-T. Yau High School Science Award