Morfismos, Vol. 1, No. 1, 1997, pp. 35–46

The complexity of coding problems

Irasema Sarmiento[1]

Abstract

In this work we deal with two problems related with the weight enumerator of a linear code. That is, determining the middle coefficient and the number of vectors in the code with no zero entries. We prove that the first problem is NP-hard because determining any coefficient of the weight enumerator is Turing reducible to determining the middle one. We prove as well that the second problem is NP-complete by reducing it to the problem of whether or not a graph is 3-colourable. We include the necessary background in complexity and matroids.

1991 Mathematics Subject Classification: 94B05, 68Q25.
Keywords and phrases: *Complexity, Linear Codes, Matroids.*

1 Introduction

Here we give an informal introduction to the complexity concepts used in the next sections.

For any finite set Σ of symbols, we denote by Σ^* the set of all finite strings of symbols from Σ. If $L \subseteq \Sigma^*$, we say that L is a *language* over the alphabet Σ. An *encoding scheme* e for a *problem* Π provides a way of describing each instance of Π by an appropriate string of symbols over some fixed alphabet Σ. The *decision problems* have only two possible solutions, yes or no.

The language that we associate with Π and e is:

$$\begin{aligned} L[\Pi, e] \quad = \quad & \{x \in \Sigma^* : \Sigma \text{ is the alphabet used by } e, \\ & \quad \text{and } x \text{ is the encoding under } e \text{ of an instance } I \in Y_\Pi\} \end{aligned}$$

[1] Ph.D. student, Merton College, Oxford.

where Y_Π is the set of "yes" instances.

A *deterministic Turing machine* (DTM) is a model of computation. A program for a DTM includes a finite set of states with two distinguished halt-states q_Y and q_N. We say that a DTM *program* M with input alphabet Σ *accepts* $x \in \Sigma^*$ if and only if M halts in a state q_Y when applied to input x. The language L_M *recognised* by the program M is given by:

$$L_M = \{x \in \Sigma^* : M \quad \text{accepts} \quad x\}.$$

We say that a DTM program M *solves* the decision problem Π under encoding scheme e if M halts for all input strings over its input alphabet and $L_M = L[\Pi, e]$.

The *time* used in the computation of a DTM program M on an input x is the number of steps occurring in that computation up until a halt state is entered. For a DTM program M that halts for all inputs $x \in \Sigma^*$, its *time complexity function* $T_M : Z^+ \to Z^+$ is given by:

$$\begin{aligned} T_M(n) \quad = \quad & \max\{m : \text{there is an } x \in \Sigma^*, \text{ with } |x| = n, \text{ such that} \\ & \qquad \text{the computation of } M \text{ on input } x \text{ takes time } m\}. \end{aligned}$$

Such a program M is called a *polynomial time* DTM program if there exists a polynomial p such that $T_M(n) \leq p(n)$ for all positive integers n.

The class of languages P is defined as:

$$\begin{aligned} P = \{L : & \text{ there is a polynomial time} \\ & \qquad \text{DTM program } M \text{ for which } L = L_M\}. \end{aligned}$$

We say that a decision problem Π belongs to P under the encoding scheme e if $L[\Pi, e] \in P$.

The class NP is intended to capture the idea of *polynomial time verifiability*, that is, given an instance I it can be verified in polynomial time if the answer for I is yes. Note that polynomial time verifiability does not imply polynomial time solvability unless $NP = P$. Formally NP can be defined using the notion of a program for a *nondeterministic*

Turing machine (NDTM). Note that $P \subseteq NP$.

A *polynomial transformation* from a language $L_1 \subseteq \Sigma_1^*$ to a language $L_2 \subseteq \Sigma_2^*$ is a function $f : \Sigma_1^* \to \Sigma_2^*$ that satisfies:

1. There is a polynomial time DTM program that computes f.

2. For all $x \in \Sigma_1^*$, $x \in L_1$ if and only if $f(x) \in L_2$.

If there is a polynomial transformation from L_1 to L_2 we write $L_1 \propto L_2$. A language L is defined to be *NP-complete* if $L \in NP$ and $L' \propto L$ for all $L' \in NP$. Cook's major theorem is that NP-complete languages exist.

A *search problem* Π consists of a set D_Π of finite objects called *instances* and, for each instance $I \in D_\Pi$, a set $S_\Pi[I]$ of finite objects called *solutions* for I. An algorithm is said to *solve* a search problem Π if, given as input any instance $I \in D_\Pi$, it returns the answer "no" whenever $S_\Pi[I]$ is empty and otherwise some solution s belonging to $S_\Pi[I]$.

A *polynomial time Turing reduction* (or simply *Turing reduction*) from a search problem Π to a search problem Π' is an algorithm A that solves Π by using a hypothetical subroutine S for solving Π' such that, if S was a polynomial time algorithm for Π', then A would be a polynomial time algorithm for Π. We say that Π is Turing reducible to Π'. This can be defined formally using *oracle Turing machines*.

2 Linear codes

In this section we introduce some elementary concepts about linear codes.

We use F_q to denote the finite field with q elements, for q a prime power. A linear $[n, k]$ q-ary code C is a subspace of dimension k of a vector space V of dimension n over F_q. The members of C are called codewords. We assume that $V = F_q^n$.

Let $c = (c_1, \cdots, c_n), c' = (c'_1, \cdots, c'_n) \in C$. The *Hamming* distance between c and c' is defined as $d(c, c') = |\{i : c_i \neq c'_i\}|$; the weight of c is $w(c) = d(c, 0)$.

The *weight enumerator* of C is the polynomial $A(C, q, z) = \sum_{i=0}^{n} a_i z^i$, where $a_i = |\{c \in C : w(c) = i\}|$. Note that $a_0 = 1$.

A *generating matrix* for C is a $k \times n$ matrix over F_q such that its rows form a basis of C. The *dual* code C^* of C is $C^* = \{v \in V : v \cdot c = 0 \ \forall c \in C\}$. A generating matrix for C^* is called a *parity check matrix* for C.

Two codes are called *equivalent* if one can be obtained from the other by a sequence of operations of the following type:

(A) permutation of the positions of the code;

(B) multiplication of the symbols appearing in a fixed position by a non-zero scalar.

Let M_1 and M_2 be two generating matrices for the q-ary codes C_1 and C_2. Then these two codes are equivalent if and only if M_2 can be obtained from M_1 by a sequence of the following operations:

(R1) permutation of the rows;

(R2) multiplication of a row by a non-zero scalar;

(R3) addition of a scalar multiple of one row to another;

(C1) permutation of the columns;

(C2) multiplication of any column by a non-zero scalar.

Since equivalent codes have the same parameters (n, k) and the same weight enumerator, we can assume that, for a given code C, its generating matrix is in the *standard form* $[I_k|A]$. If it is not the case, then we can transform the given generating matrix (or a matrix whose rows are a generating set for C) into the generating matrix of an equivalent code by a sequence of the given operations and (R4) elimination of a zero row. Note that we can do it in polynomial time (this by Gaussian elimination). On the other hand, if $G = [I_k|A]$ is a generator matrix for an $[n, k]$-code C, then a parity-check matrix for C is $H = [-A^T|I_{n-k}]$, which we can obtain from G in polynomial time.

3 The hardest coefficient of $A(C, q, z)$

In this section we prove that for any $i \in \{1, \cdots, n\}$, determining a_i is Turing reducible to determining $a_{\lfloor n/2 \rfloor}$.

Let $C \subseteq F_q^m$ be an $[m, r]$-code. Let $C' = \{c' \in F_q^{m+1} : c' = (c, 0), c \in C\}$. Note that $A(C, q, z) = A(C', q, z)$ and, in fact, C and C' are isomorphic. Hence we can assume without loss of generality that m is even. Let $m/2 \leq i \leq m$ and $C'' = C \times \{0\}^{2i-m}$, then C'' is a code of length $n = 2i$ over F_q, which can be constructed in polynomial time from C and $a''_i = a''_{n/2} = |\{c'' \in C'' : w(c'') = i = n/2\}| = |\{c \in C : w(c) = i\}| = a_i$. Now, let $1 \leq i < m/2$ and let U be a generating matrix for C, $(n = 4m)$. Consider $C' \subseteq F_q^m$ with generating matrix $U' \in F_q^{r \times n}$ defined by:

$$U' = \begin{pmatrix} & 1 & 1 & \dots & 1 \\ & 0 & 0 & \dots & 0 \\ U & \vdots & \vdots & & \vdots \\ & 0 & 0 & \dots & 0 \end{pmatrix}$$

By the definition of a generating matrix, the rows $u_1, \cdots, u_r$ of U are a basis for C and

$$\forall c \in C : \exists!\ \alpha_1, \cdots, \alpha_r \in F_q : c = \alpha_1 u_1 + \cdots + \alpha_r u_r.$$

We are constructing C' in order to count the number of codewords such that $\alpha_1 \neq 0$. Note that we can construct C' in polynomial time. Observe that $\forall c' \in C' \exists!\ \alpha_1, \cdots, \alpha_r \in F_q$:

$$\begin{aligned} c' &= \alpha_1 u'_1 + \cdots + \alpha_r u'_r \\ &= \alpha_1(u_{11}, \cdots, u_{1m}, 1, \cdots, 1) + \cdots + \alpha_r(u_{r1}, \cdots, u_{rm}, 0, \cdots, 0) \\ &= (\alpha_1 u_{11} + \cdots + \alpha_r u_{r1}, \cdots, \alpha_1 u_{1m} + \cdots + \alpha_r u_{rm}, \alpha_1, \cdots, \alpha_1), \end{aligned}$$

where $u'_1, \cdots, u'_r$ are the rows of U'.

Let $\Psi : C \to C'$ be such that $\Psi(\sum_{j=1}^r \alpha_j u_j) = \sum_{j=1}^r \alpha_j u'_j$. Then Ψ is an isomorphism and the sets $\{c \in C : \alpha_1 \neq 0\}$ and $\{c' \in C' : \alpha_1 \neq 0\}$ have the same cardinality, $|\{c' \in C' : w(c') \geq n - m\}|$, because $\alpha_1 \neq 0 \Leftrightarrow w(\Psi(c)) \geq n - m$ for any $c \in C$.

Now note that the function $\Phi : \{c \in C : w(c) = i \text{ and } \alpha_1 \neq 0\} \to \{c' \in C' : w(c') = i + n - m\}$ defined by $\Phi(\sum_{j=1}^r \alpha_j u_j) = \sum_{j=1}^r \alpha_j u'_j$ is a bijection, and $|\{c \in C : w(c) = i \text{ and } \alpha_1 \neq 0\}| = |\{c' \in C' : w(c') = i + n - m\}| := \delta_1$.

But since $n/2 \leq i + n - m$, we can determine δ_1 in time bounded by a polynomial in $n = 4m$, and so, by a polynomial in m calling an oracle for $a_{\lfloor n/2 \rfloor}$.

Since $|\{c \in C : w(c) = i\}| = |\{c \in C : w(c) = i \text{ and } \alpha_1 = 0\}| + \delta_1$, we need to compute now $|\{c \in C : w(c) = i \text{ and } \alpha_1 = 0\}|$. In order to do this we define $C_1 \subseteq F_q^m$ with generating matrix

$$\begin{pmatrix} u_{21} & \dots & u_{2m} \\ \vdots & & \vdots \\ u_{r1} & \dots & u_{rm} \end{pmatrix}$$

Clearly we can construct C_1 in polynomial time from C. Continuing with this process we have $a_i = |\{c \in C : w(c) = i \text{ and } \alpha_1 = \cdots \alpha_{r-1} = 0\}| + \delta_1 + \cdots + \delta_{r-1}$, where each of $\delta_1, \cdots \delta_{r-1}$ can be determined in time bounded by a polynomial in m calling an oracle for $a_{\lfloor n/2 \rfloor}$.

On the other hand $\{c \in C : w(c) = i \text{ and } \alpha_1 = \cdots \alpha_{r-1} = 0\} = \{\alpha u_r : \alpha \in F_q \text{ and } w(c) = i\} = \{\alpha u_r : \alpha \in F_q^* \text{ and } w(c) = i\}$, which is equal to ϕ if $w(u_r) \neq i$ and $q - 1$ otherwise. Therefore we can determine a_i in polynomial time using the algorithm to determine $a_{\lfloor n/2 \rfloor}$ as a subroutine.

4 An NP-complete problem

In this section we prove that given an $[m, k]$ q-ary code C, the decision problem: is $a_m \neq 0$?, is NP-complete. In fact, we prove that the problem is NP-complete for $q = 3$, and so, we have the result for the general case. In Section 6 we give another proof of this result.

Note that when C is a binary code, we can determine a_m in polynomial time, because the only vector of length m in F_2^m is $(1, \dots, 1)$ (the all-one vector). But, working as in the case $q = 3$, the decision problem, is $a_m \neq 0$? is NP-complete for every q-ary code with $q > 2$. Also, for every fixed prime power q and positive integer t, we can determine $\{a_0, a_1, \dots, a_t\}$ in polynomial time using exhaustive search.

Let $G = (V, E)$ be a loop less connected graph with $V = \{v_1, \dots, v_n\}$ and $E = \{e_1, \dots, e_m\}$, and for all $j = 1, \dots, m : e_j = \{v_{j_1}, v_{j_2}\}$ with $j_1 < j_2$. A *proper (vertex) 3-colouring* of G is a function $\phi : V \to F_3$ such that $\phi(u) \neq \phi(v)$ if $\{u, v\} \in E$. If such a function exists we say that G is *3-colourable*. It is well known that the problem of deciding whether or not G is 3-colourable is NP-complete.
We construct in polynomial time a code $C < F_3^m$ such that $a_m \neq 0$ if and only if G is 3-colourable. That is, we prove that the decision problem, is G 3-colourable? is polynomial reducible to, is $a_m \neq 0$?.
Let $W = \{\phi : V \to F_3 : \phi \text{ is a function }\}$. Then W is a F_3 vector space with the usual operations.
For $i = 1, \dots, n$ we define $\chi_{v_i} : V \to F_3$ such that $\chi_{v_i}(v) = 1$ if $v = v_i$ and 0 otherwise; that is, χ_{v_i} is the characteristic function of $\{v_i\}$.
Then $B = \{\chi_{v_1}, \dots, \chi_{v_n}\}$ is a basis of W.
Now note that the elements of W are precisely the 3-colourings of G. For every $\phi \in W$ we define $c_\phi = (c_{\phi_1}, \dots, c_{\phi_m}) \in F_3^m$ such that for all $j = 1, \dots, m,\ c_{\phi_j} = \phi(v_{j_2}) - \phi(v_{j_1})$. Define $C = \{c \in F_3^m : \exists \phi \in W : c = c_\phi\}$. Then C is a subspace of F_3^m and $a_m \neq 0$ if and only if G is 3-colourable. In fact:

(i) $c_\phi + c_\psi = c_{\phi+\psi}$

(ii) $\forall \alpha \in F_3 : \alpha c_\phi = c_{\alpha\psi}$.

From this we can easily see that $C = \langle c_{\chi_{v_1}}, \dots, c_{\chi_{v_n}} \rangle$. Here $St(v) = \{e \in E(G) : v \text{ is incident with } e\}$, and note that for all $i = 1, \dots, n$ and for all $j = 1, \dots, m,\ c_{(\chi_{v_i})j}$ is equal to 0 if e_j does not belong to $St(v_i)$, -1 if $e_j \in St(v_i)$ and $v_i = v_{j_1}$, and 1 if $e_j \in St(v_i)$ and $v_i = v_{j_2}$.

Given G we can construct these vectors in polynomial time. Let M be a matrix whose rows are $R_1, \dots, R_n$ and such that $R_i = c_{\chi_{v_i}}$ for all $i = 1, \dots, n$. Using Gauss-elimination we can construct in polynomial time from M a generating matrix for our code C. So, the decision problem, is $a_m \neq 0$? is NP-complete.

5 Matroids and its representations

A *matroid* M is a pair $(E, \mathcal{I})$, where E is a finite set and $\mathcal{I}$ is a collection of subsets of E (the *independent* sets of M) satisfying the following conditions:

(I1) $\phi \in I$.

(I2) If $I_1 \in I$ and $I_2 \subseteq I_1$, then $I_2 \in I$.

(I3) If I_1 and I_2 are in I and $|I_1| < |I_2|$, then $\exists x \in I_2 - I_1 : I_1 \cup x \in I$.

A subset of E that is not in $\mathcal{I}$ is called *dependent*.

The following are important subsets of the ground set $E(M)$ of a matroid M:

(i) The set of *circuits* of M which are the minimal dependent sets.

(ii) The set of *bases* of M which consists of the maximal independent sets.

The *rank* of $A \subseteq E(M)$ is the cardinality of a maximal independent set contained in A.

A matroid can be defined by circuits, bases, or rank as well as by independent sets. The *dual* of M, denoted by M^*, is a matroid with ground set $E(M)$ and bases set $\{E(M) - B : B \text{ is a basis of } M\}$. A *loop* is a circuit of M with one element, and a *coloop* (or *isthmus*) is a co-circuit of M (that is, a circuit of M^*) with cardinality one.

If $T \subseteq E(M)$, there is a matroid $M \setminus T$ (called the *deletion of* T *from* M) on $E \setminus T$ whose independent sets are those independent sets in M that are contained in $E \setminus T$. The *contraction of* T *from* M is defined by $M/T = (M^* \setminus T)^*$.

If E is the set of edges of a graph G and $\mathcal{I}$ is the set of forests of G, then $\mathcal{I}$ is the set of independent sets of a matroid $M(G)$ on E called the *cycle matroid* of G.

Two matroids $M = (E, \mathcal{I})$ and $M' = (E', \mathcal{I}')$ are said to be *isomorphic* if there exists a bijection $\phi : E \to E'$ such that $I_1 \in \mathcal{I}$ if and only if $\phi(I_1) \in \mathcal{I}'$.

Let $m = |E|$, F be a field and A be an $r \times m$ matrix over F. The columns of A span a subspace W of F^r and form a matroid M' where $\mathcal{I}$ is defined by linear independence. If M is isomorphic to M' we say that A is a *representation* of M over F. For example, given a graph G (with n vertices and m edges) and a field F, the matrix A constructed as follows is a representation of $M(G)$ over F. We orient the graph G in the following way, let $e_j = \{v_{j_1}, v_{j_2}\}$ $(j_1 < j_2)$, then (v_{j_2}, v_{j_1}) be the directed edge. Now consider the matrix $A \in F_3(n \times m)$ such that $a_{ij} = 1$ if the vertex i is the tail of arc j, -1 if vertex i is the head of arc j and 0 otherwise; and reduce each entry of A modulo F.

A matroid representable over F_2 is called *binary*, a matroid representable over F_3 is called *ternary* and a matroid representable over every field is called *regular*.

We prove now that the code C defined in Section 4 is generated by the rows of a matrix representation of $M(G)$ over the field F_3. Here $M(G)$ is the cycle matroid of the graph G. Let $M \in F_3(n \times m)$ such that for all $i = 1, \ldots, n$, R_i (the ith row of M) is $C_{\chi_{v_i}}$. Then $m_{ij} = (c_{\chi_{v_i}})j = 1$ if $e_j \in St(v_i)$ and $v_i = v_{j_2}$, -1 if $e_j \in St(v_i)$ and $v_i = v_{j_1}$, and 0 otherwise; that is $m_{ij} = 1$ if v_i is the tail of the arc j, -1 if v_i is the head of the arc j, and 0 otherwise. Therefore $A = M$. It is known that A is a representation of $M(G)$ over F_3.

6 The Tutte polynomial and the weight enumerator of a linear code

Let M be a matroid with ground set E and rank function r, we define its *Tutte polynomial* as $t(M; x, y) = \sum_{X \subseteq E} (x-1)^{r(E)-r(X)} (y-1)^{|X|-r(X)}$. This is unique because of the following theorem.

Theorem 6.1 *There is a unique function from the set of isomorphism classes of matroids to the polynomial ring $\mathbb{Z}[x, y]$ having the properties:*

(i) $t(I; x, y) = x$ *(I denotes an isthmus).*

(ii) $t(L; x, y) = y$ *(L denotes a loop).*

(iii) *If $e \in E(M)$, then (deletion-contraction)*

(a) $t(M; x, y) = t(M \setminus e; x, y) + t(M/e; x, y)$ *if e is neither a loop nor an isthmus;*

(b) $t(M;x,y) = xt(M \setminus e;x,y)$ *if* e *is an isthmus;*

(c) $t(M;x,y) = yt(M/e;x,y)$ *if* e *is a loop.*

Let M_1 and M_2 be two matroids with independent sets $\mathcal{I}_1$ and $\mathcal{I}_2$ respectively. Assume that $E(M_1) \cap E(M_2) = \phi$. Then the direct sum of M_1 and M_2 is the matroid with ground set $E_1 \cup E_2$ and independent sets $\{I_1 \cup I_2 : I_1 \in \mathcal{I}_1, I_2 \in \mathcal{I}_2\}$; this matroid is denoted by $M_1 \bigoplus M_2$.

A *Tutte-Grothendieck invariant* is a function f defined on a class of matroids closed under minors which satisfies:

(i) $f(M) = af(M \setminus e;x,y) + bf(M/e;x,y)$ for $e \in E(M)$ not a loop or an isthmus.

(ii) $f(M_1 \bigoplus M_2) = f(M_1)f(M_2)$.

Theorem 6.2 *If* f *is a Tutte-Grothendieck invariant then*
$f(M) = a^{|E|-r(E)}b^{r(E)}t(M;\frac{x_0}{b},\frac{y_0}{a})$
where x_0 *and* y_0 *are the values* f *takes on coloops and loops respectively.*

Given a linear code C and a generating matrix A of C, the matroid on the columns of A (defined by linear independence) depends only on C and not on the choice of A; this matroid is denoted by $M(C)$. Let C^* be the dual code of C, then $M(C^*)$ is isomorphic to $M^*(C)$.

If A is a representation of the matroid M over some finite field F_q, then we denote by $C(M)$ its associated linear code, the row space of A.

Concerning the weight enumerator we have the following

Proposition 6.3 $A(C;q,z) = (1-z)^k z^{n-k} t(M(C);\frac{1+(q-1)z}{1-z},\frac{1}{z})$.

The classical MacWilliams duality formula for linear codes can be proved using this proposition. The MacWilliams formula is

$$A(C^*;q,z) = \frac{(1-(q-1)z)^n}{q^k} A(C;q,\frac{1-z}{1+(q-1)z}).$$

We already proved that if C is an $[n,k,d]$ q-ary code, then the decision problem: is $a_n \neq 0$?, is NP-complete. We give here another proof of this fact using the Tutte polynomial.

Let G be a graph. Let A be the matrix representation of $M(G)$ over F_q constructed as described in Section 5, we can find it in polynomial time. Given a colouring ψ of G, we say that $e = \{u, v\} \in E(G)$ is a *bad edge* if $\psi(u) = \psi(v)$. The *bad colouring polynomial* of G is $B(G; q, z) = \sum_{l=0}^{n} b_l(q) z^l$ where $b_l(q)$ is the number of q colourings of G with exactly l bad edges.

It is known that $B(G; q, z) = (z-1)^k q t(M(G); \frac{z-1+q}{z-1}, z)$ where k is the rank of $M(G)$. Now let $C = C(M(G))$. Then $A(C; q, z) = (1-z)^k z^{n-k} t(M(G); \frac{1+(q-1)Z}{1-z}, \frac{1}{z})$ and

$$\begin{aligned}
\frac{z^n}{q} B(G; q, z^{-1}) &= z^n(\frac{1}{z} - 1)^k t\left(M(G); \frac{(1/z) - 1 + q}{(1/z) - 1}, \frac{1}{z}\right) \\
&= z^n(\frac{1-z}{z})^k t\left(M(G); \frac{1 - z + qz}{1-z}, \frac{1}{z}\right) \\
&= (1-z)^k z^{n-k} t\left(M(G); \frac{1 + (q-1)z}{1-z}, \frac{1}{z}\right) \\
&= A(C; q, z).
\end{aligned}$$

Therefore $A(C; q, z) = \frac{z^n}{q} B(G; q, \frac{1}{z})$, so

$$\sum_{i=0}^{n} a_i z^i = \frac{z^n}{q} \sum_{l=0}^{n} b_l(q) z^{-l} = \sum_{l=0}^{n} \frac{b_l(q)}{q} z^{n-l} = \sum_{i=0}^{n} \frac{b_{n-i}(q)}{q} z^i.$$

Then for all $i = 0, \cdots, n : a_i = \frac{b_{n-i}(q)}{q}$ implies that for all $i = 0, \cdots, n : q a_i = b_{n-i}(q)$ which is the number of q-colourings of G with $n - i$ bad edges. Thus $q a_n = b_0(q)$ which is the number of good colourings of G.

For example, with $q = 3$, we have that $a_n \neq 0 \Leftrightarrow G$ is 3-colourable. So the problem: is $a_n \neq 0$? is NP-complete. As a corollary we have that determining a_n is $\sharp P$-hard.

Acknowledgment

I thank my advisor D.J.A. Welsh for his support and guidance. I am grateful to the CONSEJO NACIONAL DE CIENCIA Y TECNOLO-

GIA (CONACyT) of México for the financial support I am receiving during my D.Phil. studies at Oxford.

Irasema Sarmiento
Merton College,
Oxford OX1 4JD,
UK
sarmient@maths.ox.ac.uk

References

[1] Garey M. R., Johnson D. S. *Computers and intractability*, W.H. Freeman and Company, 1991.

[2] Oxley J. G. *Matroid theory*, Oxford University Press, 1992.

[3] MacWilliams F. J., Sloane N. J. A. *The theory of error-correcting codes* North Holland, 1983.

[4] Hill R. *A first course in coding theory*, Oxford University Press, 1993.

[5] Brylawski T., Oxley J. G. *The Tutte polynomial and its applications*, Cambridge University Press, 1992.

[6] Welsh D. J. A. *Complexity: knots, colourings and counting*, Cambridge University Press, 1993.