Two-Parts of Unlabeled Tournament Numbers

S. C. Cater and R. W. Robinson
Computer Science Department
415 GSRC
University of Georgia
Athens, GA 30602

Abstract

Let $t(n)$ be the number of unlabeled tournaments on n nodes, let $v_2(i)$ denote the 2-part of i, and let $E(n) = v_2(t(n)) - \lfloor n/2 \rfloor$. In a previous paper the authors showed that $E(n) = 0$ for all odd n, proved that $E(n) \geq 0$ for all even $n \geq 4$, and characterized even n for which $E(n) = 0$. In the present paper the values of even n for which $E(n) = 1, 2$ and 3 are characterized. The characterizations lead to randomized algorithms for determining when $E(n) = 0, 1, 2,$ and 3 which are sublinear in time complexity as a function of n. Also a deterministic algorithm is presented for computing $E(n)$ whenever $E(n) \leq k$, for which the running time is polynomial in n provided k is fixed as $n \to \infty$.

1 Introduction

Let $t(n)$ be the number of unlabeled tournaments on n nodes, let $\varphi(n)$ denote the Euler totient function, and let $v_2(n)$ be the 2-part of n, i.e., the exponent of the longest power of 2 which divides n exactly. Let

$$E(n) = v_2(t(n)) - \lfloor n/2 \rfloor,$$

which we term the *excess 2-part* (for tournament numbers) at n. In [2] it is shown that $E(n) \geq 0$ for $n \geq 3$, with equality if, and only if, either n is odd, or else n is even and $\varphi(n)/2$ is odd. In the same paper a similar result is obtained for the numbers of unlabeled graphs.

In the present paper our attention is restricted to tournament numbers and even $n \geq 4$. We will characterize the values of n for which $E(n) = 1, 2$, and 3. We will also discuss the derivation of algorithms for computing $E(n)$ which have allowed most values for n up to 7680 to be determined. The data are consistent with the hypothesis that $E(n)$ is bounded, and in fact with $E(n) \leq 16$; however there is no known theoretical justification for such a conjecture.

We repeat such notation and facts from [2] as is necessary for the presentation to be logically self-contained. We use $\sigma \vdash n$ to signify that σ is a numerical partition of n. Two conventions are needed for specifying partitions. One is to simply list the parts, as

$$\sigma = [s_1, s_2, \ldots, s_k].$$

Here the order in which the parts are listed is immaterial, $s_i \geq 0$ for each $i = 1, \ldots, k$, and if $\sigma \vdash n$ then the parts sum to n, i.e., $n = \sum_1^k s_i$. Also $k = k(\sigma)$ denotes the number of parts in σ from now on. The other convention is to list in order the number of parts of each size, as

$$\sigma = \langle \sigma_1, \sigma_2, \ldots, \sigma_n \rangle.$$

Here σ_i denotes the number of parts which are equal to i, for $i = 1, \ldots, n$. Thus $\sigma_i \geq 0$ for each $i = 1, \ldots, n$ and if $\sigma \vdash n$ then $n = \sum_1^n i\sigma_i$. For tournaments we can restrict attention to *odd* partitions, that is, partitions in which every part is odd. We use $\sigma \models n$ to signify that σ is an odd partition of n. Then $t(n)$ can be expressed as

$$t(n) \;=\; 2^{n/2} \sum_{\sigma \models n} \Delta(\sigma), \tag{1}$$

where

$$\begin{aligned} \Delta(\sigma) &= 2^{\Lambda(\sigma)}/\Omega(\sigma), \\ \Lambda(\sigma) &= -\frac{k}{2} + \sum_{1 \leq i < j \leq k} (s_i, s_j), \\ \Omega(\sigma) &= \prod_{i=1}^{n} \sigma_i! i^{\sigma_i}, \end{aligned} \tag{2}$$

and (s_i, s_j) denotes the greatest common divisor (g.c.d.) of s_i and s_j. As explained in [2], the formula for $t(n)$ was first published by Davis [3], and can also be found in [5, §5.2] or [7, §29]. In (2) we have made use of a minor simplification in expressing $\Lambda(\sigma)$ using $k = k(\sigma)$, as was done in [2, §1]. The only difference is that the term $n/2$ has been removed from $\Lambda(\sigma)$ in the present paper.

2 Polynomial Algorithm for Bounded Excess 2-Part

Fix $M \geq 0$, and consider the problem of computing the function E_M defined by

$$E_M(n) = \begin{cases} E(n) & \text{if } E(n) \leq M, \\ M+1 & \text{if } E(n) > M. \end{cases}$$

This is equivalent to computing excess 2-parts for tournament numbers which are bounded by M. We will show that for fixed M, $E_M(n)$ can be computed in time polynomial in n. We will first derive a crude bound from which this follows. We will then refine the bounds with a view to improving computational efficiency and to paving the way for a sublinear (in n) algorithm for E_3.

The key to efficient computation of E_M is to limit attention to only those odd partitions of n which might affect the result. ¿From [2] we know that $E(n) = E_M(n) = 0$ whenever n is odd, so from now on we restrict consideration to even n. If n is even and $\sigma \models n$ we define the *excess 2-part of* σ to be

$$E(\sigma) = v_2(2^{n/2}\Delta(\sigma)) - \frac{n}{2} = v_2(\Delta(\sigma)).$$

Then from (1) we have

$$E_M(n) = \overline{v_2}\Bigg(\sum_{\substack{\sigma \models n \\ E(\sigma) \leq M}} \Delta(\sigma) \Bigg), \tag{3}$$

where $\overline{v_2}(i) = \min\{v_2(i), M+1\}$ for all i. This is because the addition of terms with excess 2-part greater than M can have no effect on the excess 2-part of the sum if the latter is not greater than M.

Since n is even and σ is an odd partition of n, the number $k = k(\sigma)$ of parts of σ is even. Also $n \geq 2$, so $k \geq 2$. For an odd partition the powers in $\Omega(\sigma)$ do not contribute to the 2-part, so from (2) we have

$$E(\sigma) = \sum_{1 \leq i < j \leq k} (s_i, s_j) - \frac{k}{2} - \sum_{1 \leq l \leq n} v_2(\sigma_l!). \tag{4}$$

An obvious lower bound for the sum over the g.c.d.'s in (4) is $k(k-1)/2$, since each g.c.d. is at least 1. For bounding the other sum in (4), it will be helpful to note that $v_2(p!) \leq p-1$ for all $p \geq 1$, and that $v_2(p!) \leq p-2$ for all odd $p \geq 3$. These follow immediately from the more general fact that $v_2(p!)$ is $p - c(p)$ for all integers $p \geq 0$, where $c(p)$ is the number of 1's in the binary representation of p. Since $\sum \sigma_l = k$, this gives the crude bound

$$\sum_{1 \leq l \leq n} v_2(\sigma_l!) \leq k-1. \tag{5}$$

Thus in all (4) gives the bound

$$E(\sigma) \geq \frac{k(k-1)}{2} - \frac{k}{2} - (k-1) = f(k) \tag{6}$$

where

$$f(k) = \frac{k^2}{2} - 2k + 1.$$

Now for any M let $k(M)$ be given by

$$k(M) = \min\left\{k : k \geq 2 \ \&\ k \text{ is even } \&\ f(k+2) > M\right\}.$$

Since $f(k)$ is monotone increasing for $k > 2$, and only even values of k can occur, we know that $E(\sigma) > M$ whenever $k(\sigma) > k(M)$. Thus from (3) we have the following

Theorem 1 *For even $n \geq 2$ and $M \geq 0$,*

$$E_M(n) = \overline{v_2}\Big(\sum_{\substack{\sigma \models n \\ k(\sigma) \leq k(M)}} \Delta(\sigma)\Big).$$

The theorem gives an algorithm of time complexity $O(n^{k(M)-1})$ for computing $E_M(n)$, or of time complexity $O(n^{k(M)})$ for computing the full sequence of values $\langle E_M(2i) : 1 \leq i \leq n/2\rangle$. The reason is that there are $O(n^{k-1})$ partitions of n into at most k parts for any $k \geq 1$, and the same is true for partitions into odd parts (with smaller implied constant corresponding to O). To see this in a completely elementary way, recall that there are exactly $\binom{n+k-1}{k-1}$ ordered numerical partitions of n into k parts. On the one hand, this is an upper bound for the number of unordered partitions which occur in Theorem 1. On the other hand, for fixed k this bound is $O(n^{k-1})$.

In order to refine our bounds and improve our algorithm for computing E_M, we will consider the number of different parts sizes $s(\sigma)$ in a partition σ. We let

$$S(\sigma) = \{i : \sigma_i > 0\},$$

so that $s(\sigma) = |S(\sigma)|$. If $s(\sigma) = s$ and $k(\sigma) = k$ then in the sum $\sum v_2(\sigma_l!)$ there are s occurrences of the bound $v_2(\sigma_i) \leq \sigma_i - 1$, one for each i in $S(\sigma)$. Thus

$$\sum_{l=1}^{n} v_2(\sigma_l!) \leq k - s \tag{7}$$

can replace (5). Our object is to improve the lower bound for $E(\sigma)$ to $f(k)+3$, which we obtain at once from (7) when $s(\sigma) \geq 4$.

To attain the improved lower bound for $E(\sigma)$ in case $s(\sigma) \leq 3$, we will need to exclude a class A_n of odd partitions of n, which will need to be considered separately as special cases. For even $n \geq 2$, let

$$\begin{aligned} A_n = \{\sigma \models n \ :\ & (S(\sigma) = \{1\}) \text{ or } (S(\sigma) = \{1, n-k+1\} \ \&\ \sigma_1 = k(\sigma) - 1) \\ & \text{or } (S(\sigma) = \{1, a, b\} \ \&\ \sigma_1 = k(\sigma) - 2 \ \&\ (a, b) = 1)\}. \end{aligned}$$

Then for $\sigma \notin A_n$, we can replace $k(M)$ with $k(M-3)$ in Theorem 1, leading to the following refinement.

Theorem 2 *For even $n \geq 2$ and $M \geq 0$,*

$$E_M(n) = \overline{v_2}\left(\sum_{\sigma \models n}{}' \Delta(\sigma)\right)$$

where $\sum'$ denotes the restriction to $\{\sigma : \sigma \in A_n \text{ or } k(\sigma) \leq k(M-3)\}$.

To finish verifying Theorem 2, we need to show that $\Delta(\sigma)$ has no effect on $E_M(n)$ when $s(\sigma) = 1, 2$, or 3, $\sigma \notin A_n$, and $k(\sigma) > k(M-3)$.

Case 1. $s(\sigma) = 1$.

Since $S(\sigma) \neq \{1\}$, we have $S(\sigma) = \{a\}$ for some $a \geq 3$. Since $k(\sigma) > k(M-3)$ (where $k = k(\sigma)$), we have $f(k)+3 > M$. Each of the g.c.d.'s in (4) is a, and therefore is at least 3. So (4) gives

$$E(\sigma) \ \geq\ \frac{3k(k-1)}{2} - \frac{k}{2} - (k-1) \ =\ g(k)$$

where

$$g(k) \ =\ \frac{3k^2}{2} - 3k + 1.$$

It is readily verified that $g(k) \geq f(k)+3$ for all integers $k > 2$. But $k > k(M-3) \geq 2$, so in fact $E(\sigma) > M$.

Case 2. $s(\sigma) = 2$.

By (7) we can improve (6) to $E(\sigma) \geq f(k)+1$ in this case, using only that each g.c.d. in (4) is at least 1. Since k is even and $k > k'(M) \geq 2$, we have $k \geq 4$. And since $\sigma \notin A_n$, it must be the case that $\sigma_a \geq 2$ for some $a > 1$. Since a is odd, $a \geq 3$. We now have at least 2 parts of size a, giving a g.c.d. of a instead of 1. Therefore we can increase our lower bound by $a - 1 \geq 2$, so that $E(\sigma) \geq f(k) + 3$. But then $E(\sigma) > M$ follows as in Case 1.

Case 3. $s(\sigma) = 3$.

By (7) we can improve (6) to $E(\sigma) \geq f(k) + 2$ in this case, using only that each g.c.d. in (4) is at least 1. As in Case 2 we have $k \geq 4$, so some part occurs at least twice. Also as in Case 2 we have $E(\sigma) > M$ if some part $a > 1$ occurs at least twice, so we may assume that only the part 1 is repeated. Thus $\sigma \in A_n$ unless $S(\sigma) = \{1, a, b\}$ where $(a, b) \neq 1$. But then again $(a, b) \geq 3$ since a and b are odd, and our lower bound can be increased by $(a, b) - 1 \geq 2$. Thus $E(\sigma) \geq f(k) + 4 > M$.

3 Sublinear Algorithm for E_3

We will characterize the values of n for which $E(n) = 0, 1, 2$ and 3 (individually) in terms of the function $v_2(\varphi(n)/2)$. We will denote the latter by $\psi(n)$ since it occurs so often. These characterizations will give an algorithm for E_3 which has the same order of time-complexity as $\psi(n)$. Note that we consider complexity as a function of n, rather than $\log n$.

Theorem 3 *For even $n \geq 4$,*

$$E(n) = i \text{ iff } \psi(n) = i, \text{ for } i = 0, 1, \text{ and } 2, \text{ and}$$

$$E(n) = 3 \text{ iff } \begin{cases} \psi(n) = 3 & \text{and } \psi(n-2) > 0 \\ & \text{or} \\ \psi(n) > 3 & \text{and } \ \psi(n-2) = 0. \end{cases}$$

Proof In [2] we showed that $E(n) = 0$ if, and only if, $\psi(n) = 0$. As $\psi(4) = 0 = \psi(6)$, we assume $n \geq 8$ for the rest of the proof. In order to treat $E(n) = i$ for $i = 1, 2$, and 3 we will need the following fact, which has been proved by Andrew J. Granville [4].

Lemma 1 *For even $n \geq 4$ let*

$$R(n) = \sum_{\substack{1 \leq a < n/2 \\ (a,n)=1}} \frac{1}{a(n-a)};$$

then $v_2(R(n)) = \psi(n)$.

Note that the bound on k used in Theorem 2, $k(M-3)$, takes the value 2 for $M \leq 3$. Thus apart from partitions in the class A_n, we only need consider the contributions to $t(n)2^{-n/2}$ in equation (1) which arise from partitions of n into two odd parts, say $\sigma = [a, n-a]$. Since $(a, n-a) = (a, n)$ we can express the total of these contributions as $W_2(n) + U(n)$, where

$$W_2(n) = \sum_{\substack{1 \leq a < \frac{n}{2} \\ a \text{ odd}}} \frac{2^{(a,n)-1}}{a(n-a)},$$

$$U(n) = \begin{cases} 0 & \text{if } n/2 \text{ is even,} \\ \frac{2^{n/2-1}}{2(n/2)^2} & \text{if } n/2 \text{ is odd.} \end{cases}$$

The term $U(n)$ is the contribution of the partition $[n/2, n/2]$. When $U(n) \neq 0$, $v_2(U(n)) = n/2 - 2$ and so $U(n)$ will not affect $E_3(n)$ as long as $n \geq 12$. But $8/2$ is even, and we shall see later that $E(10) = 1$, so in fact, we can disregard $U(n)$ over the whole range of values under consideration.

The value of (a, n) must be an odd divisor of n. Collecting terms with equal g.c.d.'s, we have

$$W_2(n) = \sum_{\substack{d|n \\ d \text{ odd}}} \sum_{\substack{1 \leq a < n/2 \\ (a,n)=d}} \frac{2^{d-1}}{a(n-a)} = \sum_{\substack{d|n \\ d \text{ odd}}} \frac{2^{d-1}R(n/d)}{d^2}. \tag{8}$$

If $n/2$ is odd, the term $R(2)$ corresponding to $d = n/2$ is zero. For odd $d < n/2$ we have $v_2(R(n/d)) = \psi(n/d)$ by Granville's Lemma, so the 2-part of the corresponding summand in (8) is $d - 1 + \psi(n/d)$. For $n \geq 8$, $R(n) > 0$ and we have

$$v_2(W_2(n)) = \psi(n). \tag{9}$$

That is, the summand for $d = 1$ is positive and its 2-part determines the 2-part of the whole sum. To see that the summand for odd d in the range $1 < d < n/2$ cannot alter the 2-part of the sum, note that $\psi(n) - \psi(n/d) \leq v_2(\varphi(d)) \leq \log_2(d-1) < d-1$, so $\psi(n) < d - 1 + \psi(n/d)$.

Returning to $\sigma \in A_n$, the proof of Theorem 1 shows that $E(\sigma) > 3$ whenever $k(\sigma) > k(3) = 4$. Thus we only need consider $\sigma \in A_n$ with $k(\sigma) = 4$, as the cases when $k(\sigma) = 2$ are already taken account of in (8). Now the first option in A_n, that $S(\sigma) = \{1\}$, implies that $\sigma = [1, 1, 1, 1]$, contradicting our assumptions that $\sigma \vdash n$ and $n \geq 8$. The second and third options give $\sigma = [1, 1, 1, n-3]$ and $\sigma = [1, 1, a, n-2-a]$, where $1 < a < (n-2)/2$ and $(a, n-2) = 1$. In either case, $E(\sigma) = 3$. In all there are precisely $\varphi(n-2)/2$ of these terms, which is an odd number if, and only if, $\psi(n-2) = 0$. Letting

$$W_4(n) = \sum_{\sigma \models n}{}' \Delta(\sigma)$$

where Σ' denotes the restriction to $\{\sigma : \sigma \in A_n \;\&\; k(\sigma) = 4\}$, we thus have

$$v_2(W_4(n)) \begin{cases} = 3 \text{ if } \psi(n-2) = 0, \\ > 3 \text{ if } \psi(n-2) > 0, \end{cases}$$

for $n \geq 8$.

In extracting all terms of possible significance to E_3 using Theorems 1 and 2, we have shown that

$$t(n)2^{-n/2} = W_2(n) + W_4(n) + X(n)$$

where $v_2(X(n)) \geq 4$. Thus for $n \geq 8$,

$$E_3(n) = \overline{v_2}(W_2(n) + W_4(n)),$$

where $\overline{v_2}(i) = \min\{i, 4\}$. Further, since $v_2(W_4(n)) \geq 3$, $E_2(n)$ is determined just by $v_2(W_2(n))$, which is $\psi(n)$ by (9). This proves the theorem when $i \leq 2$.

Finally, suppose $E(n) \geq 3$. Then as we have just seen, $\psi(n) \geq 3$. We have $v_2(W_2(n)) = \psi(n) \geq 3$, and $v_2(W_4(n))$ either equal to 3 or greater than 3 according to whether $\psi(n-2)$ is equal to 0 or greater than 0. Thus $v_2(W_2(n) + W_4(n)) = 3$, and hence $E(n) = 3$, when precisely one of these two conditions is an equality. □

Theorem 3 provides $E_3(n)$ in terms of $\psi(n)$ and possibly $\psi(n-2)$, so the order of the time complexity of computing $E_3(n)$ is bounded by that of $\psi(n)$. In turn, $\psi(n)$ is computed directly from $\varphi(n)$, and $\varphi(n)$ is readily determined from the prime factorization of n. Let

$$L(n) = e^{\sqrt{\log n \log\log n}}.$$

There is a randomized factoring algorithm with rigorous time complexity $L(n)^{\sqrt{4/3}+o(1)}$ which appeared recently [9], and this is improved to $L(n)^{1+o(1)}$ in a forthcoming paper [6]. As a function of n these complexity bounds are sublinear.

We know of no way to compute E_3, E_2, or E_1 in general without having to factor n. However, E_0 can be computed without factorization, in time $O(\log^c n)$ for some constant c. In this special case we are able to replace factorization by a test for primality. The well-known Solovay-Strassen test recognizes composite numbers in random time which is polynomial in $\log n$ [8]. More recently a complementary test has been devised which recognizes prime numbers and is in the same complexity class [1]. Taken together, these tests provide a randomized algorithm which will determine whether n be prime or composite in time $O(\log^c n)$ for a suitable fixed c.

Here is how to decide whether or not $\psi(n) = 0$ for even $n \geq 4$, based on primality determination. First, if $v_2(n) \geq 3$ then $\psi(n) \geq 1$. If $v_2(n) = 2$, then $\psi(n) = 0$ if, and only if, $n = 4$. If $v_2(n) = 1$, then $n = 2q$ for odd $q > 1$ and $\psi(n) = v_2(\varphi(q)) - 1$. In this case, $v_2(\varphi(q)) = 1$ if, and only if, $q = p^u$ for some $u \geq 1$ and prime $p \equiv 3 \pmod 4$. To test for this, we find the integer pair (x, t) with t as large as possible so that $x^t = q$. Since $x \geq 3$, $t \leq \log_3 q$, so there are only $O(\log n)$ powers to try. For each t it is straightforward to decide whether the real t-th root of q is an integer in time $O(\log^4 n)$ uniformly in t. We now check to see if $x \equiv 3 \pmod 4$, and if so apply the randomized algorithm of [1] and [8] to determine whether x is prime.

4 Work in Progress and Open Problems

Based on the ideas presented in Section 2, algorithms have been developed for computing $E(n)$ in a manner which is much more efficient than computing $t(n)2^{-n/2}$ and then extracting the 2-part. Whereas the authors were only able to report on $E(n)$ for $n < 100$ based on the latter approach in [2], computations based on the efficiencies introduced in the present paper have allowed the determination of $E(n)$ for all $n \leq 7640$, except for the two values $n = 5472$ and 7590. The data are consistent with the hypothesis that $E(n) \leq 16$ for all n. However it is an open problem to prove this, or indeed to prove that $E(n)$ is bounded as $n \to \infty$.

In light of the sublinear algorithm for E_3 given in Section 3, it seems reasonable to seek an algorithm for E_4 which is not much harder. At present, however, the best algorithm known to the authors is the one provided by Theorem 2, which computes a single value $E_4(n)$ in time $O(n^3)$. It is an open problem to find an algorithm for E_4 which is asymptotically more efficient than that.

Acknowledgement: The authors thank Carl Pomerance for making a number of helpful suggestions.

References

[1] L. Adleman and M.-D. A. Huang, Recognizing primes in random polynomial time, *Proc. 19th ACM Symp. on Theory of Computing* (1987) 462–469.

[2] S. C. Cater and R. W. Robinson, Exponents of 2 in the numbers of unlabeled graphs and tournaments, *Congr. Numer.* **82** (1991) 139–155.

[3] R. L. Davis, Structures of dominance relations, *Bull. Math. Biophys.* **16** (1954) 131–140.

[4] A. J. Granville, private communication, 1991.

[5] F. Harary and E. M. Palmer, *Graphical Enumeration*, Academic, New York, 1973.

[6] H. W. Lenstra, Jr. and Carl Pomerance, A rigorous time bound for factoring integers, *J. Amer. Math. Soc.*, to appear.

[7] J. W. Moon, *Topics on Tournaments*, Holt, New York, 1968.

[8] R. Solovay and V. Strassen, A fast Monte-Carlo test for primality, *SIAM J. Comput.* **6** (1977) 84–85; Erratum **7** (1978) 118.

[9] B. Vallée, Generation of elements with small modular squares and provably fast factoring algorithms, *Math. Comp.* **56** (1991) 823–849.