Refined Upper and Lower Bounds for 2-SUM

Alexander C. Chan * William I. Gasarch† Clyde P. Kruskal‡
University of Maryland University of Maryland University of Maryland

Abstract

We prove upper and lower bounds on the time complexity of solving the 2-SUM problem: given a set of numbers, are there two of them that sum to zero? Our basic models are the linear decision tree and the degree-d algebraic decision tree. Our bounds are more precise than is common for this field and allow us to observe that 2-SUM is strictly harder than sorting in the linear decision tree model.

1 Introduction, Upper Bound, and Model

The 3-SUM problem is: Given n numbers, do any *three* of them sum to zero? This problem is important in Computational Geometry [BBG94, Eri96, GO95, ORou94] because if 3-SUM requires $\Omega(n^2)$ steps (which is likely [Eri96, ES95]) then the following problems (and others) also require $\Omega(n^2)$ steps:

1. Given n points in the plane, determine if some three of them that are co-linear [GO95].

2. Given n triangles, compute the area of their union [GO95].

We study a related problem, namely 2-SUM: Given n numbers, do any *two* of them sum to zero? We first show that this problem can be solved with $O(n \log n)$ linear comparisons and requires $\Omega(n \log n)$ queries on a d-ADT (see definition below). We refine the constants on both the upper and lower bounds. The mathematics used for the refined lower bound is of interest and may be useful on other problems such as Element Distintness and Two-list Element Distinctness (discussed in Section 6.) In addition our results show that 2-SUM is *strictly harder* than sorting in the linear decision tree model, which is clearly of interest.

To clarify our model we exhibit a well-known upper bound. First sort the numbers in nondecreasing order. Let $i = 1$, $j = n$, and $s := x_i + x_j$. Test $s \leq 0$. If $s \leq 0$ then we need to test $s \geq 0$, otherwise we do not. If $s = 0$ then we are done. if $s < 0$ then $i = i + 1$, and if $s > 0$ then $j := j - 1$. In either case $s = x_i + x_j$ and repeat until $j < i$. In effect we are keeping pointers at each end of the sorted list and moving them toward each other in a fashion that guarantees that if there are two numbers that sum to zero, we will find them. Otherwise we determine that no two (distinct) numbers sum to zero. The entire algorithm requires sorting and then at most $2n - 2$ comparisons. Since the initial sorting can be done in $n \lg n - 1.329n$ comparisons [FJ59, HL69, Knu73, Man79], this yields an *upper bound* of

*Dept. of C.S., U. of MD, College Park, MD 20742. (alexchan@cs.umd.edu).
†Dept. of C.S. and Inst. for Adv. Comp. Studies, U. of MD, College Park, MD 20742. Supported in part by NSF grant CCR-97-32692 (gasarch@cs.umd.edu).
‡Dept. of C.S. U. of MD, College Park, MD 20742. (kruskal@cs.umd.edu).

$n \lg n - 1.329n + 2n \leq n \lg n + 0.67n$ comparisons. We will give an improved upper bound later in this paper.

The algorithm above uses comparisons and questions of the form "$x + y = 0$?" The latter is viewed as asking "$x + y \leq 0$?" and "$-x - y \leq 0$?" Hence the natural basic operation to consider is the 2-ary linear comparisons: questions of the form "$ax + by \leq c$?". A more general model would allow questions of the form "$\sum_{i=1}^{n} a_i x_i$ COMP b?" where COMP is one of $\{<, \leq, =, \geq, >\}$. In this notation x_i is an input value and a_i and b are constants. An even more general model would allow, for some fixed d, a comparison between a polynomial in $x_1, \ldots, x_n$ of degree d and a constant.

A sequential algorithm involving linear comparisons is represented as a *linear decision tree*, (henceforth an LDT) a finite rooted binary tree with a linear comparison at each node and YES and NO edges from each node to its children. The input $(x_1, \ldots, x_n)$ to an LDT determines a path from the root (first linear comparison) to a leaf by the outcome of the linear comparisons encountered. The answer at the leaf node is the output of the LDT. A sequential algorithm involving comparisons between polynomials of degree d and constants is called an *algebraic decision tree of degree* d (henceforth a d-ADT) and is defined similarly.

If T is an LDT or d-ADT then let $\mathrm{ht}(T)$ be the height of the tree, i.e., the maximum number of internal nodes on a path from the root to a leaf. In Section 4 we show that if T is a d-ADT for 2-SUM then $\mathrm{ht}(T) \geq \Omega(n \log n)$. In Section 5 we refine this lower bound.

Our final results are

1. 2-SUM can be computed with $n \lg n + 0.351n + O(\lg n)$ linear queries.
2. 2-SUM requires $n \lg n - 0.92n - \Omega(\lg n)$ linear queries.
3. 2-SUM requires $\frac{0.38n \lg n - 0.96n}{d} - \Omega(\lg n)$ queries on a d-ADT.

2 Preliminaries

The following proposition, due to Dobkin and Lipton [DL79] and Ben-Or [Ben83] can be used to obtain lower bounds.

Definition 2.1 A set $X \subseteq R^n$ is *connected* if for all points $x, y \in X$ there is a path from x to y that is entirely inside X. Let $\mathcal{E} \subseteq R^n$. X is a *connected component of* $\mathcal{E}$ if $X \subseteq \mathcal{E}$, X is connected, and no superset of X is contained in $\mathcal{E}$.

Proposition 2.2 *Let $\mathcal{E}$ be the union of N connected components in R^n.*

1. *[DL79] If T is an LDT for determining membership in $\mathcal{E}$ then*

$$\mathrm{ht}(T) \geq \lg N.$$

2. *[Ben83] If T is a d-ADT for determining membership in $\mathcal{E}$ then*

$$\mathrm{ht}(T) \geq \frac{0.38 \lg N - 0.61n}{d}.$$

(This is obtained by looking at Ben-Or's paper more carefully than is commonly done. The constant 0.38 is a close upper bound for $\frac{1}{1+\lg 3}$. The constant 0.61 is a close lower bound for $\frac{\lg 3}{1+\lg 3}$.)

We would like to cast 2-SUM as a decision problem in this framework. Let

$$\mathcal{E}_n = \{(x_1, \ldots, x_n) \in R^n : (\forall i \neq j)[x_i + x_j \neq 0]\}.$$

Clearly $\mathcal{E}_n \subseteq R^n$ is the set of all inputs to 2-SUM which answer NO. Hence we can obtain a lower bound on 2-SUM by counting the connected components of $\mathcal{E}_n$. Let $\#\mathcal{E}_n$ denote this number.

3 A Refined Upper Bound

Theorem 3.1 *There is an algorithm for 2-SUM that takes $n \lg n + 0.351n + O(\lg n)$ comparisons.*

Proof: The basic idea is to partition the numbers into two sets, the negative and non-negative numbers, and look for a number in the larger set whose additive complement is in the smaller set. The only minor hitch is that we need to check for the special case of two occurrences of 0 in the set of nonnegative numbers.

Let $S = \{a_1, a_2, \ldots, a_n\}$. Compare all numbers to 0, putting them into two sets: L, those less than 0, and G, those greater than or equal 0. Let X be the set with fewer elements, and let x be its size. Sort X, which takes time at most $x \lg x - \alpha x + \frac{1}{2} \lg x + \beta$, where $\alpha = 2 - \lg 3 + \lg e - \lg \lg e \approx 1.329$ and $\beta < 3.3$ [FJ59, HL69, Knu73, Man79].

If $X = G$ then check if the list starts with two 0's; if so, we are done. If $X = L$ append a 0 to the end of L. For every element $a \in S - X$ look for $-a$ in X. (It takes two matches to succeed for 0.) The searches take at most $(n - x)(\lceil \lg(x + 1) + 1 \rceil)$ comparisons using binary search.

The total number of comparison steps for this algorithm is at most

$$\begin{aligned} & n + x \lg x - \alpha x + \frac{1}{2} \lg x + \beta + 2 + (n - x)(\lceil \lg(x + 1) + 1 \rceil) \\ \leq \quad & n + x \lg x + \frac{1}{2} \lg(x + 1) - \alpha x + \beta + 2 + (n - x)(\lg(x + 1) + 2) \\ \leq \quad & (n + \frac{1}{2}) \lg(x + 1) - (2 + \alpha)x + 3n + \beta + 2 \qquad (1) \end{aligned}$$

To maximize, take the derivative and set to 0.

$$\frac{(n + \frac{1}{2}) \lg e}{x + 1} - (2 + \alpha) = 0 \implies x = \frac{(n + \frac{1}{2}) \lg e}{2 + \alpha} - 1$$

Substituting back into (1) the total number of steps is at most

$$\begin{aligned} & (n + \frac{1}{2}) \lg(\frac{(n + \frac{1}{2}) \lg e}{2 + \alpha}) - (2 + \alpha) \left(\frac{(n + \frac{1}{2}) \lg e}{2 + \alpha} - 1 \right) + 3n + \beta + 2 \\ = \quad & n \lg(n + \frac{1}{2}) + \frac{1}{2} \lg(n + \frac{1}{2}) + (3 - \lg e - \lg(2 + \alpha) + \lg \lg e)n - \alpha + \beta \\ \leq \quad & n \lg n + 0.351n + \frac{1}{2} \lg n + O(1) \end{aligned}$$

∎

4 An Easy Lower Bound

The following lower bound, while easy, does not seem to be in the literature.

Theorem 4.1 *If T is a d-ADT for $\mathcal{E}_n$ then* $\mathrm{ht}(T) \geq \Omega(n \log n)$.

Proof: We assume n is even. The case of n odd is similar. Let $n = 2m$. We show that $\mathcal{E}_n$ has at least $m! = \Omega(n \log n)$ connected components.

Let $\sigma \in S_m$. Let A_σ be the set of all $(x_1, \ldots, x_m, y_1, \ldots, y_m)$ such that

$$-x_m < y_{\sigma(1)} < -x_{m-1} < y_{\sigma(2)} < \cdots < -x_1 < y_{\sigma(m)} < 0 < x_1 < x_2 < \cdots < x_m.$$

It is easy to see that each A_σ is a connected component of $\mathcal{E}_n$. Since there are $m!$ such components we are done. ∎

5 A Refined Lower Bound

The set $\mathcal{E}_n$ can be expressed as a disjoint union of nonempty open sets. The number of these open sets is the number of connected components of $\mathcal{E}_n$. We call these open sets *the cells of* $\mathcal{E}_n$. We need to count them.

To this end we introduce (undirected) threshold graphs. Our goal is to establish a bijection between the cells of $\mathcal{E}_n$ and the labeled threshold graphs on n vertices. Since threshold graphs have been enumerated, we have a count of the cells of $\mathcal{E}_n$.

Definition 5.1 [Stan] A *threshold graph* may be defined recursively as follows:

1. The empty graph is a threshold graph.
2. If G is a threshold graph, then so is the disjoint union of G with a one-vertex graph.
3. If G is a threshold graph, then so is the (edge) complement of G.
4. No other graph is a threshold graph.

Note 5.2 If graph G has an isolated vertex v, then G is a threshold graph iff $G - v$ is a threshold graph, by Definition 5.1, condition 2.

Notation 5.3 We denote the number of threshold graphs on n vertices by $t(n)$.

Notation 5.4 If $G = (V, E)$ is a graph and $v \in V$ then $G - \{v\}$ is the graph

$$(V - \{v\}, E - \{\{u, v\} : u \in V\}).$$

Lemma 5.5 *If $n \geq 2$ then $\#\mathcal{E}_n = t(n)$.*

Proof: Let $I = \{(i,j) : 1 \le i < j \le n\}$. Let E_n be a mapping from I to $\{<,>\}$. Throughout the proof we view E_n as an unordered set of inequalities of the form $x_i + x_j < 0$ or $x_i + x_j > 0$. For each E_n let O^{E_n} be the open set $\{(x_1,\ldots,x_n) : (x_i + x_j)\ E_n[i,j]\ 0\}$. Note that $\mathcal{E}_n$ is the disjoint union of O^{E_n}'s over all possible E_n's. Unfortunately many of the O^{E_n} are empty. We map all the E_n such that $O^{E_n} \neq \emptyset$ onto the set of all threshold graphs on n vertices.

We define a function $\mathbf{P}$ that will, given a map E_n such that $O^{E_n} \neq \emptyset$, return a threshold graph on n vertices. Given E_n such that $O^{E_n} \neq \emptyset$, let $\mathbf{P}(G)$ be the graph on n vertices that has edge (i,j) iff $E_n[i,j]$ is $>$. We show that $\mathbf{P}$ is 1-1 and maps onto the set of threshold graphs.

Range of $\mathbf{P}$ *is Threshold Graphs:* For $n = 2$ this is easy. Suppose (by way of contradiction) that $n > 2$ is the smallest value such that there exists an E_n such that $O^{E_n} \neq \emptyset$ and $\mathbf{P}(E_n)$ is not a threshold graph. We show that n is not minimal. $\mathbf{P}(E_n)$ does not have an isolated vertex: assume it had an isolated vertex i. Then E_n thinks $x_i + x_j < 0$ for all j. Hence if E'_{n-1} is E_n without any of the inequalities that mention x_i then $O^{E'_{n-1}} \neq \emptyset$ and $\mathbf{P}(E'_{n-1})$ is not a threshold graph, contradicting the minimality of n. Similarly, the complement of $\mathbf{P}(E_n)$ cannot have an isolated vertex. Let $(x_1,\ldots,x_n) \in O^{E_n}$. Let $i\text{-}\min$ be such that $x_{i\text{-}\min} = \min_{1\le i\le n} x_i$. Let $i\text{-}\max$ be such that $x_{i\text{-}\max} = \max_{1\le i\le n} x_i$.

If $x_{i\text{-}\min} + x_{i\text{-}\max} > 0$ then $x_{i\text{-}\max} + x_i > 0$ for all $1 \le i \le n$ and $i\text{-}\max$ is an isolated vertex in the complement of $\mathbf{P}(E_n)$. If $x_{i\text{-}\min} + x_{i\text{-}\max} < 0$ then $x_{i\text{-}\min} + x_i < 0$ for all $1 \le i \le n$ and $i\text{-}\min$ is an isolated vertex in $\mathbf{P}(E_n)$. So $x_{i\text{-}\min} + x_{i\text{-}\max} = 0$ which implies $(x_1,\ldots,x_n) \notin \mathcal{E}_n$. Contradiction.

One-to-one: If $E_n \neq E'_n$ then they differ on some (i,j). Hence the graphs $\mathbf{P}(E_n)$ and $\mathbf{P}(E'_n)$ differ on (i,j)

Onto: The $n = 2$ case is easy. Suppose for $n \ge 2$, $\mathbf{P}$ maps onto threshold graphs on n vertices. Consider G, a threshold graph on $n+1$ vertices. Either G or $\overline{G}$ has an isolated vertex i such that $G - \{i\}$ is a threshold graph. We assume it is G, the other case is similar. Renumber so that $i = n+1$. Let E_n be such that $\mathbf{P}(E_n) = G - \{n+1\}$ and $O^{E_n} \neq \emptyset$. Let E'_{n+1} be $E_n \cup \{x_{n+1} + x_i > 0 : 1 \le i \le n\}$. Clearly $O^{E'_{n+1}} \neq \emptyset$ and $\mathbf{P}(E'_{n+1}) = G$.

The bijection establishes the desired result. ∎

We now need a good approximation for $t(n)$.

Theorem 5.6 $t(0) = t(1) = 1$. $(\forall n \ge 2)[t(n) = 2 + \sum_{i=2}^{n-1} \binom{n}{i} t(i)]$.

Proof: Let $s(n)$ denote the number of threshold graphs with no isolated vertex. Since a threshold graph is a choice of isolated vertices, $I \subseteq \{1,\ldots,n\}$, plus a threshold graph with no isolated vertex on the remaining vertices, $\{1,\ldots,n\} - I$, $t(n) = \sum_{i=0}^{n} \binom{n}{i} s(i)$. Observe from Definition 5.1 that a threshold graph with $n \ge 2$ vertices has no isolated vertex if and only if its complement has an isolated vertex. Hence $t(n) = 2s(n)$ for $n \ge 2$. For $n < 2$ we calculate $s(1) = 0$ and $s(0) = t(0) = t(1) = 1$. Combining the two equations for $t(n)$ we obtain, for $n \ge 2$,

$$\begin{aligned} t(n) &= \binom{n}{0}s(0) + \binom{n}{1}s(1) + \binom{n}{n}s(n) + \textstyle\sum_{i=2}^{n-1}\binom{n}{i}s(i) \\ t(n) &= 1 + 0 + \frac{t(n)}{2} + \frac{1}{2}\textstyle\sum_{i=2}^{n-1}\binom{n}{i}t(i) \\ 2t(n) &= 2 + t(n) + \textstyle\sum_{i=2}^{n-1}\binom{n}{i}t(i) \\ t(n) &= 2 + \textstyle\sum_{i=2}^{n-1}\binom{n}{i}t(i) \end{aligned}$$

∎

We need to estimate $t(n)$. To do this we need the following lemma which easily follows from the Remainder theorem for Taylor series. We include an elementary proof for simplicity and completeness.

Lemma 5.7 *For all $s \in R$ and $s \in N$, $\sum_{i=1}^{n-2} \frac{s^i}{i!} \geq e^s - 1 - \frac{s^{n-1}}{(n-1)!} e^s$ and $\sum_{i=1}^{n-2} \frac{s^i}{i!} \leq e^s - 1$*

Proof:

$$\begin{aligned}
\textstyle\sum_{i=1}^{n-2} \frac{s^i}{i!} &= e^s - 1 - \textstyle\sum_{i=n-1}^{\infty} \frac{s^i}{i!} \\
&= e^s - 1 - \frac{s^{n-1}}{(n-1)!} \textstyle\sum_{i=n-1}^{\infty} \frac{s^{i-n+1}(n-1)!}{i!} \\
&= e^s - 1 - \frac{s^{n-1}}{(n-1)!} \textstyle\sum_{j=0}^{\infty} \frac{s^j (n-1)!}{(n+j-1)!} \\
&\geq e^s - 1 - \frac{s^{n-1}}{(n-1)!} \textstyle\sum_{j=0}^{\infty} \frac{s^j}{j!} \\
&= e^s - 1 - \frac{s^{n-1}}{(n-1)!} e^s
\end{aligned}$$

$$\sum_{i=1}^{n-2} \frac{s^i}{i!} = e^s - 1 - \sum_{i=n-1}^{\infty} \frac{s^i}{i!} \leq e^s - 1.$$

▮

Theorem 5.8

1. $t(n) = O(\frac{n!}{(\ln 2)^n})$.
2. $\lg t(n) = n \lg n - n \lg(\ln 2) + \Theta(\lg n) \geq n \lg n - 0.92n$.

Proof: We prove that, for all $n \geq 2$, $t(n) \geq a\frac{n!}{s^n} + bn$ by constructive (mathematical) induction, deriving values for the constants a, b and s.
Base cases: To satisfy the $n = 2$ cases we need the following constraint on a, b, s

$$a\frac{2}{s^2} + 2b \leq 2;$$

Induction step: We may assume that $n \geq 3$ and that the inequality is true for all natural numbers less than n. Then

$$\begin{aligned}
t(n) &= 2 + \sum_{i=2}^{n-1} \binom{n}{i} t(i) \\
&\geq 2 + \sum_{i=2}^{n-1} \binom{n}{i} \left[a\frac{i!}{s^i} + bi\right] \qquad \text{by the induction hypothesis} \\
&= 2 + a \sum_{i=2}^{n-1} \binom{n}{i} \frac{i!}{s^i} + b \sum_{i=2}^{n-1} \binom{n}{i} i \\
&= 2 + a \sum_{i=2}^{n-1} \frac{n!}{(n-i)!i!} \frac{i!}{s^i} + b \sum_{i=2}^{n-1} n \binom{n-1}{i-1} \\
&= 2 + a \sum_{i=2}^{n-1} \frac{n!}{s^i (n-i)!} + bn \sum_{i=2}^{n-1} \binom{n-1}{i-1} \\
&= 2 + a \sum_{i=1}^{n-2} \frac{n!}{s^{n-i} i!} + bn \sum_{i=1}^{n-2} \binom{n-1}{i} \\
&= 2 + a\frac{n!}{s^n} \sum_{i=1}^{n-2} \frac{s^i}{i!} + bn[2^{n-1} - 2]
\end{aligned}$$

$$\begin{aligned}
&\geq && 2 + a\frac{n!}{s^n}\left[e^s - 1 - \frac{s^{n-1}}{(n-1)!}e^s\right] + bn[2^{n-1}-2] \quad \text{by Lemma 5.7} \\
&= && 2 + a\frac{n!}{s^n}\left[e^s - 1\right] - ae^s\frac{n}{s} + bn[2^{n-1}-2] \\
&= && 2 + a\frac{n!}{s^n} - 2a\frac{n}{s} + bn[2^{n-1}-2] \quad \text{setting } e^s - 1 = 1, \text{ or } s = \ln 2 \sim 0.69
\end{aligned}$$

We need a, b such that

$$2 + a\frac{n!}{s^n} - 2a\frac{n}{s} + bn[2^{n-1}-2] \geq a\frac{n!}{s^n} + bn$$

hence

$$2 - 2a\frac{n}{s} + bn[2^{n-1}-3] \geq 0$$

This is hardest to satisfy when n is small. Since $n \geq 3$ we only have to satisfy the $n = 3$ case which is

$$2 - \frac{6a}{s} + 3b \geq 0.$$

Arithmetic shows that $a = \frac{s^2}{3} \sim 0.16$ and $b = 0.6$. will satisfy this constraint and the one from the base case.

We prove that, for all $n \geq 2$, $t(n) \leq a\frac{n!}{s^n} + bn$ by constructive (mathematical) induction, deriving values for the constants a, b and s.

Base cases: To satisfy the $n = 2$ cases we need the following constraint on a, b, s

$$a\frac{2}{s^2} + 2b \geq 2;$$

Induction step: We may assume that $n \geq 3$ and that the inequality is true for all natural numbers less than n. Then

$$\begin{aligned}
t(n) &\leq 2 + a\frac{n!}{s^n}\sum_{i=1}^{n-2}\frac{s^i}{i!} + bn[2^{n-1}-2] \quad \text{similar to algebra in other case} \\
&\leq 2 + a\frac{n!}{s^n}\left[e^s - 1\right] + bn[2^{n-1}-2] \quad \text{by Lemma 5.7} \\
&= 2 + a\frac{n!}{s^n} + bn[2^{n-1}-2] \quad \text{setting } e^s - 1 = 1, \text{ or } s = \ln 2 \sim 0.69
\end{aligned}$$

We need b such that

$$2 + a\frac{n!}{s^n} + bn[2^{n-1}-2] \leq a\frac{n!}{s^n} + bn$$

hence

$$2 + bn[2^{n-1}-3] \leq 0.$$

This is hardest to satisfy when n is small (we will be taking $b < 0$). Since $n \geq 3$ we only have to satisfy the $n = 3$ case which is $2 + 3b \leq 0$. Arithmetic shows that $a = 2.5s^2 \sim 1.2$ and $b = -0.7$. will satisfy this constraint and the one from the base case.

Using Stirling's formula we get: $t(n) = \Theta\left(\sqrt{n}\left(\frac{n}{e\ln 2}\right)^n\right)$. Hence
$\lg t(n) = n\lg n - (\lg(e\ln 2))n + \Theta(\lg n) \geq n\lg n - 0.92n + \Theta(\lg n)$. ∎

Note 5.9 Computer evidence seems to indicate that $t(n) = 0.442695(\frac{n!}{(\ln 2)^n} + O(1)$. Note that 0.442695 looks suspiciously like $(\ln 2)^{-1} - 1$.

Theorem 5.10

1. *If T is an LDT that solves 2-SUM then*

$$\text{ht}(T) \geq n \log n - 0.92n + 0.5 \log n + \Omega(1).$$

2. *If T is a d-ADT that solves 2-SUM then*

$$\text{ht}(T) \geq \frac{0.38n \lg n - 0.96n + 0.19 \lg n}{d}.$$

Proof: This follows from Proposition 2.2 and Theorem 5.8. ∎

Corollary 5.11 *On the LDT model 2-SUM is strictly harder than sorting.*

Proof: By Theorem 5.10 2-SUM requires at least $n \lg n - 0.92n + 0.5 \lg n - 2.30$. Sorting can be done with $n \lg n - 1.329n$ comparisons [FJ59, HL69, Knu73, Man79]. The conclusion follows. ∎

6 The Two List Element Distinctness Problem

The algorithm in Theorem 3.1 can be phrased informally as (1) split the list into two groups X and Y, and (2) see if an element of X is the negation of some element of Y. We can study part (2) in isolation.

Definition 6.1 The *two list element distinctness problem* (TLED henceforth) is the problem of determining membership in

$$TLED_{p,q} = \{[X = (x_1, \ldots, x_p)], [Y = (y_1, \ldots, y_p)] : (\forall i, j)[x_i \neq y_j]\}.$$

Theorem 6.2 *The TLED problem can be solved in $n \lg n - 0.64n + O(\log n)$ comparisons.*

Proof sketch: Assume $p \leq q$. Sort X. For every element of Y look for it in X via binary search. We omit the algebra. ∎

We have a partial result on lower bounds. We try to show that $TLED_{p,q}$ can be expressed as the disjoint union of a certain number of nonempty open sets. We have that number as a recurrence, but not in closed form.

If A and B are sets then $A < B$ means that every element of A is less than every element of B. Consider the set

$$\{(x_1, x_2, x_3, x_4, y_1, y_2, y_3, y_4) : \{x_2, x_4\} < \{y_3, y_4\} < \{x_1\} < \{y_1, y_2\} < \{x_3\}\}.$$

This is an open subset of $TLED_{4,4}$. More generally, $TLED_{p,q}$ can be partitioned into disjoint nonempty open set of this type. Let $c(p, q)$ be the number of such sets that begin with an a subset of $\{x_1, \ldots, x_p\}$. It is easy to show that $c(1, q) = 1$, $c(p, 0) = 1$, and $c(p, q) = \sum_{i=1}^{p} \binom{p}{i} c(q, p - i)$. With some manipulation one can obtain $c(p, q) = \sum_{i=0}^{p-1} \binom{p}{i} c(i + 1, q - 1)$. Using these recurrences and Propostion 2.2 we obtain that a computable lower bound for $TLED_{p,q}$ is $\lg(c(p, q) + c(q, p))$.

7 Open Problems

Several open problems suggest themselves.

1. Obtain tighter upper and lower bounds for 2-SUM for both the LDT and d-ADT models.

2. The *Element Distinctness Problem* (henceforth ED) is the following: given $(x_1, \ldots, x_n)$ determine if there exists $i \neq j$ such that $x_i = x_j$. ED can be solved in $n \lg n - 0.33n + O(1)$ comparisons (first sort then compare adjacent elements). It is know that ED requires $\Omega(n \log n)$ queries on a d-ADT and other models [Ben83, BLY92, GK94, Lop94]. It is easy to show that ED requires $\lg n! = n \lg n - 1.44n + \Omega(1)$ operations on an LDT. It is unknown how ED compares with sorting and 2-SUM. We conjecture that ED is harder than sorting but easier than 2-SUM.

3. The lower bound for 3-SUM using an LDT is $\Omega(n \lg n)$, far from the upper bound of $O(n^2)$. Jeff Erickson has proven a $\Omega(n^2)$ lower bound for 3-SUM [Eri96] if linear comparisons are restricted to be of the form "$ax_i + bx_j + cx_k$ COMP d?". Unfortunately the upper bound of $O(n^2)$ uses 4-ary linear comparisons for which his result does not apply. The open problem here is to obtain better lower bound for 3-SUM on 4-ary LDT's, LDT's, and d-ADT's.

8 Acknowledgements

Sloane's On-line Encyclopedia of Integer Sequences [Slo97] was used to discover the combinatorial relationship between 2-SUM and threshold graphs. Thanks to Richard Stanley, Clara Chan (no relation), and Christos Athanasiadis for providing enumerative combinatorics answers and references. Thanks to Jason Howald and Ashley Reiter for inspired algebra. Thanks to Samir Khuller and David Mount for entertaining the odd idea from time to time. Thanks to Andrew Lee for proofreading.

References

[Ben83] M. Ben-Or. Lower bounds for algebraic computation trees. *Proceedings of the 15th Annual ACM Symposium on the Theory of Computing*, 80-86, 1983.

[BLY92] A. Björner, L. Lovász, and Andrew C.C. Yao. Linear decision trees: Volume estimates and topological bounds. *Proceedings of the 24th Annual ACM Symposium on the Theory of Computing*, 170-177, 1992.

[BBG94] S. Block, J. Buss, and J. Goldsmith. How hard are n^2-hard problems. *SIGACT News*, 25(2):83–85, 1994.

[DL79] David P. Dobkin and Richard J. Lipton. On the complexity of computations under varying sets of primitives. *Journal of Computer and System Sciences*, 18:86-91, 1979.

[Eri96] Jeffrey G. Erickson. *Lower Bounds for Fundamental Geometric Problems.* Ph.D. Dissertation, UC-Berkeley, 1996. `http://www.cs.duke.edu:80/~jeffe/pubs/thesis.html`

[ES95] J. Erickson and R. Seidel. Better lower bounds on detecting affine and spherical degeneracies. *Discrete Computational Geometry*, 13:41–57, 1995. Earlier version in FOCS93. Erratum in Disc. Comp. Geom. in 1997.

[FJ59] L. Ford and S. Johnson. A tournament problem. *American Mathematical Monthly*, 66:387-389, 1959.

[GK94] Dima Grigoriev and Marek Karpinski. Lower bound for randomized linear decision tree recognizing a union of hyperplanes in generic position. *Research Report*, 85114-CS, University of Bonn, 1994.

[GO95] Gajen and Overmars. On a class of $O(n^2)$ problems in comp. geom. *Comp. Geom. Theory Appl.*, 5:165-185, 1995.

[HL69] F. Hwang and S. Lin. An analysis of ford-johnson's sorting algorithm. In *Proceedings of the third annual Princeton Conf. on Inform. Sci. and Systems*, pages 292–296, 1969.

[Knu73] Donald E. Knuth. *Sorting and Searching,* vol. 3 of *The Art of Computer Programming.* Addison-Wesley, 1973.

[Lop94] Alex Lopez-Ortiz. New lower bounds for element distinctness on a one-tape turing machine. *Information Processing Letters*, 51:311-314, 1994. `http://daisy.uwaterloo.ca/~alopez-o/papers.html`

[Man79] G. K. Manacher. The ford-johnson sorting algorithm is not optimal. *Journal of the ACM*, 26(3):441–456, July 1979.

[ORou94] J. O'Rourke. Computational geometry column 22. *SIGACT News*, 25(1):32–33, 1994.

[Slo97] N.J.A. Sloane. *Sloane's On-Line Encyclopedia of Integer Sequences*, AT&T, 1997. `http://www.research.att.com/~njas/sequences/`

[Stan] Richard P. Stanley. *Enumerative Combinatorics*, vol. 2, Wadsworth and Brooks/Cole, due in 1998. `http://www-math.mit.edu/~rstan/ec/ec.html`