A Survey of Communication Complexity for Proving Lower Bound of Data Structures in Cell-Probe Model

Abstract

Proving lower bounds for computational problem is always a challenging work. In this survey, we will present some techniques for proving lower bound of data structures in cell-probe model. There is a natural relationship between cell-probe model and communication complexity, so many proofs of lower bound in cell-probe model are related to communication complexity. In communication complexity, there are plenty of techniques for proving communication lower bounds, which can be used directly for proving data structure lower bounds.

1 Introduction

Cell-probe model is a model for data structures first introduced by Yao [Yao81] in 1981. After that, many studies have appeared for proving lower bounds in cell probe model. Due to the essential differences between static and dynamic data structures, the studies can also be classified into these two categories.

[MNSW95] is one of the earliest paper of techniques for lower bounds of static data structures. It talks about the richness technique and the round elimination technique. Both techniques lead to a family of various techniques, and they are still the two major family of techniques today. [PT06b] introduces direct sum technique, which is a stronger richness technique and can be used for proving higher lower bounds. [PT06a] uses cell probe elimination technique to prove a strong lower bound for. Cell probe elimination is quite similar to round elimination technique, but cell probe elimination is used directly for the data structure problem, while round elimination technique is used for the corresponding communication problem. [PTW10] introduces a new cell sampling method, which is used in [Lar12b] for proving lower bound of polynomial evaluation problem. This lower bound stays as the highest lower bound of static data structures so far.

Chronogram technique introduced in [FS89] is a very influential method for proving lower bounds for dynamic data structures, which can be used to prove $\Omega(\log n/\log\log n)$ type of lower bounds. This lower bound remains the highest lower bound until [PD06], which gives the information transfer technique that can prove $\Omega(\log n)$ type of lower bounds. [Lar12a] combines chronogram technique and cell sampling to prove a $\Omega((\log n/\log\log n)^2)$ lower bound for 2-d weighted range counting problem. This type of lower bound remains highest up to date.

In this survey, we will go through most techniques that are related to communication complexity.

1.1 Cell Probe Model

In this section, we define cell probe model.

A data structure in the cell probe model has access to a set of S cells, each cell has w-bits space. This set of cells is the memory storage of the data structure. During one operation (and the first input), the data structure is assumed to have unlimited space to store any temporary information. However, if the data structure needs to store any information for further operations, it has to store it in the cells.

During each operation (and the first input), the data structure can probe cells, and are free to overwrite the cell after the data structure probes it. The data structure can choose the cells it probes based on the operation and the information in cells it has probed. If the operation is a query, the data structure needs to output the answer of the query. The update time and the query time are defined as the number of cells probed during an update operation or a query operation respectively.

Since in the cell probe, we only consider the number of memory accesses, this model is stronger than the word-RAM model, and the lower bound in this model will also hold in many other models. The lower bound in this model is always related to w and S, because if they are too large, the data structure can store all the answers during preprocessing, and thus can achieve very small cell probe complexity.

2 Static Lower bound

Static data structure problem can be defined as follows:

Definition 1. *Given a function $f : Q \times D \to \{0, 1\}$, where $D = \{0, 1\}^m$ and $Q = \{0, 1\}^n$. At first, the data structure gets the input $d \in D$, and then stores some information in the cells. Then it receives a query $q \in Q$, and it needs to output $f(q \times d)$. After the data structure gets the query q, it can only know information about d from the cells. The time used by the data structure is defined as the cells it probes after it receives the query.*

Here the definition is only for one round of query. If the data structure needs to have multiple rounds of query, the data structure must not modify the cells. If so, the time complexity for each round should be the same.

In this section, we will talk about the techniques used for proving static data structure lower bounds. Section 2.1 defines some notations and introduces the relation between static data structures and communication problem. Section 2.2 and 2.3 talk about the round elimanition technique and the richness technique introduced by [MNSW95]. Section 2.4 talks about the direct sum technique, which strengthens the idea of the richness technique. Section 2.5 introduces a generalized idea of the direct sum technique, which can be used to prove lower bounds even for two sided-error random data structures.

2.1 Preliminary

We define some notations before moving into the techniques. Let $f : \{0,1\}^n \times \{0,1\}^m \to \{0,1\}$ be a communication problem:

1. A $[a, b]$ protocol is a protocol to compute f where the total number of bits that Alice sends to Bob is at most a, and the total number of bits that Bob sends to Alice is at most b.

2. A $[t, a, b]^A$ protocol is a protocol to compute f where there are t rounds, Alice and Bob alternatively send one message in each round. Each message of Alice contains at most a bits and each round of Bob contains at most b bits. Additionally, the first message is sent by Alice. Similarly, we can define $[t, a, b]^B$ exactly the same except that Bob sends the first message.

3. A protocol is random if it outputs the correct answer with probability at least $\frac{2}{3}$. A random protocol is 1-sided error if when $f(x, y) = 0$, the protocol can output 0 with probability 1.

Static data structure relates to communication complexity in a very natural way, as we can see from the following lemma

Lemma 1. *[Mil94] Let f be a function such that there is a static data structure that can compute f in t access to the cells, with the total number of cells be S and the size of each cell be w. The communication problem defined as Alice gets $q \in Q$ and Bob gets $d \in D$, and they need to compute $f(d \times q)$, has a $[2t, \lceil \log S \rceil, w]^A$ protocol.*

Proof. We construct the protocol as follow:

1. Bob gets the input d, and he uses the data structure to construct the cells.

2. In each round, Alice requests the content of a cell by the data structure, and Bob sends the content back to Alice. Since there are at most S cells, Alice can describe the cell by $\lceil \log S \rceil$ bits; each cell has length w, so Bob can send back to Alice by w bits.

3. After t requests ($2t$ rounds), the data structure can already compute the answer, so Alice can output the answer according to the data structure.

□

Something interesting about the lemma is that it preserves some property of the data structure. If the data structure is deterministic, then the protocol is deterministic; if the data structure is randomized, then the protocol is randomized . . . Since it is usually easier to give a lower bound for the communication complexity of f, and by lemma 1 we can immediately get a lower bound for the static data structure to compute f. However, we have to note that since in $2n/\lceil \log S \rceil$ rounds, Alice can send her entire query to Bob, so the lower bound for the communication problem is at most $O(n/\lceil \log S \rceil)$. Also, since usually S has to be at least m, this shows that the technique can only prove lower bound of the type $\Omega(n/ \log m)$. Section 2.2 and 2.3 will use this transformation directly, so the techniques cannot break the barrier; whereas section 2.4 and 2.5 use the direct sum idea to prove higher lower bounds.

Miltersen et al also mention a type of converse of Lemma 1 in [MNSW95]

Lemma 2. *If there is a $[O(1), a, b]$, protocol for computing f, then there is a static data structure to compute f with $O(1)$ access to cells, where the total number of cells are $2^{O(a)}$ and the size of each cell is b.*

Lemma 2 is not focused in this survey, but it is still an interesting result. However, it requires the protocol to use only $O(1)$ rounds, and it is impossible to generalize this lemma ([MNSW95]).

2.2 The Round Elimination Technique

If $f : X \times Y \to \{0,1\}$ is a problem, then we can define $P_m(f)$ to be the following problem: Alice gets m inputs $x_1, \ldots, x_m \in X$; Bob gets integer $i \in [1, m]$, $y \in Y$ and a copy of $x_1, x_2, \ldots, x_{i-1}$ such that $P_m(f) = f(x_i, y)$ (with the inputs described). Similarly, we can define $P^m(f)$ to be the problem such that Bob gets m inputs $y_1, \ldots, y_m \in Y$; Alice gets integer $i \in [1, m]$, $x \in X$ and a copy of $y_1, y_2, \ldots, y_{i-1}$ such that $P^m(f) = f(x, y_i)$.

Lemma 3 (Round Elimination)**.** *[MNSW95] Suppose there is a randomized $[t, a, b]^A$ protocol for solving $P_{100a}(f)$. Then there is a randomized $[t-1, 120a, 120b]^B$ protocol for solving f. Similarly, if is a randomized $[t, a, b]^B$ protocol for solving $P_{100b}(f)$. Then there is a randomized $[t-1, 120a, 120b]^A$ protocol for solving f.*

The idea of the round elimination technique is described as follows: Assume there is a $[t, a, b]^A$ protocol for f, then we can use round elimination lemma repeatedly to get a 0 round protocol for a nontrivial problem. A more clear view of the technique can be seen in the following example.

Consider the following prefix *parity* problem: Let $U = \{0, \ldots, 2^n - 1\}$. Alice gets $x \in U$ and Bob gets $S \in U$ with $|S| \leq l$. They want to determine the *parity* of $|\{y \in S : y \leq x\}|$. Let the problem be $f_{n,l}$.

Theorem 1. *[MNSW95] Let $c > 1$ be a constant. Let $l = 2^{\log^2 n}$, $a = (\log n)^3$, $b = n^c$ and $t = \sqrt{\log n}/10$. Then $f_{n,l}$ does not have a $[t, a, b]^A$ protocol for sufficiently large n.*

Proof. A protocol for $f_{n,l}$ can be used for $P_m(f_{n/m,l})$ where m can be any value less than n (ignore the rounding). Alice gets $x_1, x_2, \ldots, x_m$ and she can concatenation them to get $x' = x_1 \cdot x_2 \cdots x_m$. Bob gets S and $x_1, x_2, \ldots, x_{i-1}$, so he can construct

$$S' = \{x_1 \cdot x_2 \cdots x_{i-1} \cdot u \cdot 0^{n-ni/m} : u \in S\}$$

Then they can use the protocol for $f_{n,l}$ on the input (x', S') to compute $P_m(f_{n/m,l})$. The correctness is not hard to see.

The protocol for $f_{n,l}$ can also be used for $P^m(f_{n-\log m-1}, l/m-1)$. Alice is given x and i (she does not need to use S_1 through S_{i-1}), she can compute $x' = [i-1] \cdot 0 \cdot x$ where $[i]$ represents the binary representation of i. Bob is given $S_1, \ldots, S_m$. First, he append a prefix 0 to each strings in each set, so that each set is given in length $n - \log m$. For each set S_i, if the number of elements in it is odd, we add an element $1^{n-\log m}$ to this set. Finally, compute

$$S'_i = \{[i-1] \cdot u : u \in S_i\}$$

and $S' = \bigcup_i S'_i$. Then they can use the protocol for $f_{n,l}$ on the input (x', S') to compute $P^m(f_{n-\log m-1}, l/m - 1)$. Although we add some elements to S', the total number of elements is at most $m(l/m - 1 + 1) = l$. The correctness is not hard to see.

Now, given any $[t, a, b]^A$ protocol for $f_{n,l}$, we can construct a $[t, a, b]^A$ protocol for $P_{100a}(f_{n/100a,l})$. By round elimination lemma, we have a $[t-1, 120a, 120b]^B$ protocol for $f_{n/100a,l}$. Then use the second construction, we have a $[t-1, 120a, 120b]^B$ protocol for $P^{12000b}(f_{n/100a-\log(12000b)-1,l/12000b-1})$. Use the round elimination lemma again, we have a $[t-2, 120^2a, 120^2b]^A$ protocol for

$$f_{n/100a-\log(12000b)-1,l/12000b-1}$$

By repeating the same reduction for $t/2$ times, we can get a $[0, a', b']$ protocol for asymptotic non constant n, l, which is clearly impossible. □

Corollary 1. *For any static data structure to solve prefix parity problem, if $(nl)^{O(1)}$ cells are used, and each cell contains $n^{O(1)}$ bits, then the query time is at least $\Omega(\sqrt{\log n})$.*

Proof. If the data structure probes t cells, then by lemma 1, there is a $[2t, O(\log nl), n^{O(1)}]^A$ protocol to compute prefix parity problem. When $l = 2^{\log^2 n}$, $O(\log nl) = O(\log^3 n)$ and $n^{O(1)} = n^c$ for some c, so according to theorem 1, $t \geq \Omega(\sqrt{\log n})$.

□

We can notice that prefix parity is an easier problem than the predecessor query problem, so the lower bound for prefix parity is also a lower bound for predecessor query problem. However, the lower bound here is not the strongest. Pătraşcu and Thorup proved a tight lower bound $\Omega(\log n)$ for predecessor query problem in [PT06a] via cell probe elimination. Cell probe elimination uses a quite similar idea to round elimination, but it does reduction in the original data structure problem rather than in the communication problem. It is not included in this survey, but it is a good technique for interested readers to check.

2.3 The Richness Technique

Given a function f, we can consider f as the communication matrix M, where $M_{a,b} = f(a, b)$.

Lemma 4 (Richness). *[MNSW95] We say f is (u, v)-rich if at least v columns of it contain at least u 1-entries. If f is a (u, v)-rich function and it has a randomized one-sided error $[a, b]$-protocol, then f contains a submatrix of dimensions at least $u/2^{a+2} \times v/2^{a+b+2}$ containing only 1-entries.*

Richness lemma can be used to prove lower bound of a variety of problems, here we only give one example.

Definition 2 (Disjointness Problem)**.** *Let $k < l < n/2$. Let $q \subset \{0, \dots, n-1\}$ with $|q| = k$ and $d \subset \{0, \dots, n-1\}$ with $|d| = l$. $f(q, d) = 1$ if and only if $q \cap d = \emptyset$.*

Theorem 2. *[MNSW95] If the disjointness problem has a randomized one-sided error $[a, b]$-protocol, then either $a = \Omega(k)$ or $b = \Omega(l)$.*

Proof. For any input d, we can choose q from the set that excludes d, so for any column, there are $\binom{n-l}{k}$ rows that are 1. Thus, f is $(\binom{n-l}{k}, \binom{n}{l})$-rich.

By the richness lemma, there is a $\binom{n-l}{k}/2^a \times \binom{n}{l}/2^{a+b}$ submatrix $S \times T$ of f which contains all 1. Let $t = (n-l)/2^{a/k}$. Since

$$\begin{aligned}\binom{t}{k} \Big/ \binom{n-l}{k} &\le \left(\frac{t}{n-l}\right)^k \\ &= \left(\frac{1}{2^{a/k}}\right)^k \\ &= 1/2^a\end{aligned}$$

we have

$$\binom{n-l}{k} / 2^a \ge \binom{t}{k}$$

This means $|\{i \in q : q \in S\}| \ge t$, because otherwise, we cannot have $\binom{n-l}{k}/2^a$ distinct rows. Also, because $\{i \in q : q \in S\} \cap \{i \in d : d \in S\} = \emptyset$, we have $|\{i \in d : d \in S\}| \le n - t$, which means $\binom{n}{l}/2^{a+b} \le \binom{n-t}{l}$. Therefore,

$$2^{a+b} \ge \binom{n}{l} \Big/ \binom{n-t}{l} \ge \left(\frac{n}{n-t}\right)^l \Rightarrow a + b \ge l \cdot \log \frac{n}{n-t}$$

Assume $a < k/4$. We have $t > \frac{1}{2}(n-l) > \frac{1}{4}n$ and thus $\frac{n}{n-t} > \frac{4}{3}$. Therefore,

$$b \ge l \cdot \log \frac{n}{n-t} - a \ge l \cdot \log \frac{4}{3} - \frac{l}{4} \ge \Omega(l)$$

□

Corollary 2. *For any randomized one-sided error static data structure for disjointness problem, if the number of cells used is poly(n), and the size of each cell is $O(\log n)$, then the data structure probes $\Omega(k/\log n)$ cells per query.*

Proof. If the data structure probes t cells, then by lemma 1, there is a $[2t, O(\log n), O(\log n)]$ randomized one-sided error protocol to compute the disjointness problem. The number of bits that Alice send is at most $O(t \log n)$ and the number of bits that Bob sends is also $O(t \log n)$. Then according to theorem 2, $O(t \log n) = \min(\Omega(k), \Omega(l)) = \Omega(k)$, so $t = \Omega(k/\log n)$. □

2.4 The Direct Sum Technique

As mentioned above, if we directly transform a static data structure problem f into the same communication problem f, the best lower bound we can prove is in the form $\Omega(n/\log S)$, where n

is the size of the query and S is the number of cells. However, it is still possible to use richness technique for proving stronger lower bound in the form $\Omega(n/\log\frac{Sn}{m})$ (where m is the size of the input), as we will discuss in this subsection.

Define $\bigoplus^k \mathbf{f} : ([k] \times X) \times Y^k \to \{0,1\}$ as a new problem such that the input of the data structure is $(y_1, y_2, \ldots, y_k)$, and the query consists of $x \in X$ and an index $i \in [k]$, and the data structure should output $f(x, y_i)$. We also consider another problem $\bigwedge^k f : X^k \times Y^k \to \{0,1\}$ defined as $\bigwedge^k f(x,y) = \bigcap_{i\in[k]} f(x_i, y_i)$.

Lemma 5. *[PT06b] If $\bigoplus^k f$ can be solved deterministically by a data structure in T probes, with S cells, each has w bits, then $\bigwedge^k f$ has a communication protocol in which Alice sends $O(Tk\log\frac{S}{k})$ bits and Bob sends Tkw bits.*

Proof. This protocol works in a way that computes the values of $f(x_i, y_i)$ for each i. Given the input $(x_1, x_2, \ldots, x_k)$, Alice can simulate the k subproblems in parallel. Each time she asks for k cells together, which can be encoded in $\log\binom{S}{k} = O(k\log\frac{S}{k})$ bits. Bob sends back the contents in all of the k cells by kw bits. After T rounds, Alice can figure out each $f(x_i, y_i)$, and thus she can output $\bigwedge^k f(x,y)$. □

Theorem 3. *[PT06b] Let $f : X \times Y \to \{0,1\}$ be $[\rho|X|, v]$-rich and if $\bigwedge^k f$ has a communication protocol in which Alice sends ka bits and Bob sends kb bits, then f has a 1-rectangle of size $\rho^{O(1)}|X|/2^{O(a)} \times v/2^{O(a+b)}$.*

This theorem with lemma 5 can be used to show the lower bound of $\bigoplus^k f$ in a way that is much similar to the way we use Richness Lemma 4 and lemma 1 to prove lower bound of f. One extra work is to use the lower bound of $\bigoplus^k f$ to prove the lower bound of f. One good thing about this technique is that it only concerns the communication matrix of f, and thus we can use richness and the 1-rectangle property of f as a black box. Next we will use this technique to prove the lower bound of $ANN_n^{\gamma,d}$.

Let λ be a constant. $ANN_n^{\gamma,d}$ ($\gamma \geq 1$) is a problem such that the input is n points in the space $\{0,1\}^d$. The query is a point q in $\{0,1\}^d$. If there exists a point in the input that has Hamming distant at most λ from q, then the output should be 1; if all the points in the input has Hamming distant at least $\lambda\gamma$, then the output should be 0; Otherwise, the output can be either 0 or 1. Thus, $ANN_n^{\gamma,d}$ is not actually a function, but a family of functions. In the following, we use $ANN_n^{\gamma,d}$ to denote any function that is in the family $ANN_n^{\gamma,d}$.

Theorem 4. *[PT06b] There exists a λ such that for any deterministic data structure that solves $ANN_n^{\gamma,d}$ in the cell-probe model with S cells of size $d^{O(1)}$ bits. When $d \geq (1+5\gamma)\log n$, a query requires $\Omega(\frac{d}{\gamma^3}\log\frac{Sd}{n})$ cell probes.*

Proof. If we have a data structure for $ANN_n^{\gamma,d}$ using T cell probes, then we can construct a data structure for $\bigoplus^k ANN_N^{\gamma,D}$ for some $D = d/(1+5\gamma) \geq \log n$ and $k = n/N$ for $N = w$ with the same complexity T.

We want to construct a set of binary codes of size $5\gamma D$ bits, and any two of them have Hamming distance at least γD. By Gilbert-Varshamov bound, the set of codes can have at least

$2^{(1-H(1/5)-0.01)5\gamma D} > 2^D \geq n$ elements. Given k sub problems $ANN_N^{\gamma,D}$, we can append one element in the set of codes to each sub problems. Therefore, the distance of points in two different sets is at least γD. Thus, we can store all these appended points in the data structure $ANN_n^{\gamma,d}$. If we query some point q in a sub problem and also append the code to this point, then the closest point can only be in the same sub problem; or there is no point in the same sub problem, so the closest distance is at least γD, and the answer will always be 0, as it should be when there is no point.

By lemma 5, $\bigwedge^k ANN_N^{\gamma,D}$ has a communication protocol in which Alice sends $O(Tk \log S/k)$ bits and Bob sends Tkw bits.

Then we use some result in [Liu04] as black box: there exists a λ such that the following holds,

- For any $f \in ANN_N^{\gamma,D}$, f is $[2^{D-1}, 2^{ND}]$-rich, which is equivalent to $[|X|/2, 2^{ND}]$-rich
- For any $f \in ANN_N^{\gamma,D}$, f does not contain any 1-rectangle of size $2^{D-D/(169\gamma^2)} \times 2^{ND-ND/(32\gamma^2)}$

By theorem 3, either $|X|/2^{O(T\log S/k)} < 2^{D-D/(169\gamma^2)}$, or $2^{ND}/2^{O(T\log S/k+Tw)} < 2^{ND-ND/(32\gamma^2)}$. Thus,

$$T = \Omega(\min(\frac{D}{\gamma^2}/\log\frac{S}{k}, \frac{ND}{w\gamma^2}))$$

Recall $N = w$, and the first term becomes smaller, so

$$T = \Omega(\frac{D}{\gamma^2}\log\frac{S}{k}) = \Omega(\frac{d/(1+5\gamma)}{\gamma^2}/\log\frac{S}{n/d^{O(1)}}) = \Omega(\frac{d}{\gamma^3}/\log\frac{Sd}{n})$$

□

2.5 Randomized Data Structure

All the techniques above can only be used to get lower bound for deterministic data structures or one sided error data structures. The following theorem stands as a technique for two sided error data structures:

Theorem 5. *[PT06b] Let $\alpha, \epsilon > 0$ be any constants, and $\mathbb{E}_{x\in X, y\in Y}[f(x,y)] \geq \alpha$. Assume $\bigoplus^k f$ can be solved in the cell probe model with S cells of size w in T cell probes, and error $\frac{1}{3}$, then f has a rectangle of dimensions $|X|/2^{O(T\log(S/k))} \times |Y|/2^{O(Tw)}$ in which the density of 0 is at most ϵ.*

Let NN_n^d be $ANN_n^{1,d}$. The following theorem is a direct application of theorem 5.

Theorem 6. *[PT06b] There exists a constant C such that: For any data structure that solves NN_n^d ($d \geq C\log n$) with S ($S > n$) cells of size $d^{O(1)}$ bits, and error $1/3$, a query must have $T = \Omega(d/\log\frac{Sd}{n})$ cell probes.*

Proof. First, we can use the same method as in the proof of Theorem 4 to convert a solution to NN_n^d to a solution to $\bigoplus^k NN_N^D$ for $N = n/k$ to be determined later, and $D = d/6$.

let f be the complement problem of NN_N^D. We use following results due to [BR00] as a black box:

- $\mathbb{E}_{x \in X, y \in Y}[f(x,y)] \geq \Omega(1)$
- There exists constants $\epsilon, \delta > 0$ such that any rectangle of f of size at least $|X|/O(2^{\epsilon D}/N^{\delta}) \times |Y|/2^{O(N^{\delta})}$, the density of 0 is at least 1/80.

If we use Theorem 5 here, we can obtain that f has a rectangle of dimensions $|X|/2^{O(T \log(S/k))} \times |Y|/2^{O(Tw)}$ that has at most 1/81 density of 0. Thus, we have

$$T \log \frac{S}{k} = \Omega(\epsilon D - \delta \log N) \text{ or } Tw = \Omega(N^{\delta})$$

When $d \geq C \log n$ for a large enough C, we can have

$$\epsilon D = \epsilon d/6 \geq \frac{\epsilon C}{6} \log n \geq 2\delta \log n \geq 2\delta \log N$$

Thus the first lower bound becomes $T \log \frac{S}{k} \geq \Omega(\log N) \geq \Omega(D) = \Omega(d)$. If we set N to be $(dw)^{1/\delta} = d^{O(1)}$, we have $Tw = \Omega(dw)$. Note that $\log \frac{S}{k} = \log \frac{Sd}{n} > 1$, so the first bound is stronger and thus $T = \Omega(d/\log \frac{Sd}{n})$. □

3 Conclusion

Although there have been various techniques, the limits of these techniques are still easy to see. For example, round elimination technique and the richniess technique have logorithmic dependence on the number of cells used, so it cannot give different lower bounds for a data structure that uses linear space or a data structure that uses polynomial space. The direct sum technique improves this dependence from $\log S$ to $\log(S/n)$, but can still be improved.

Also, we can see the limit of cell probe model itself. If we consider a static data structure problem, which has no input, but the query asks to compute a very hard function f. Since computing f need not to query for any cells, it has a constant cell probe complexity. However, it does not have a constant lower bound in word RAM model. It will be interesting to see a natrual data structure problem, whose upper bound in the cell probe model is lower than the lower bound in other models. Also, how large could this gap be? For such problem, it will be impossible to prove interesting lower bounds in the cell probe model.

References

[BR00] Omer Barkol and Yuval Rabani. Tighter bounds for nearest neighbor search and related problems in the cell probe model. In *Proceedings of the thirty-second annual ACM symposium on Theory of computing*, pages 388–396. ACM, 2000.

[FS89] Michael Fredman and Michael Saks. The cell probe complexity of dynamic data structures. In *Proceedings of the twenty-first annual ACM symposium on Theory of computing*, pages 345–354. ACM, 1989.

[Lar12a] Kasper Green Larsen. The cell probe complexity of dynamic range counting. In *Proceedings of the forty-fourth annual ACM symposium on Theory of computing*, pages 85–94. ACM, 2012.

[Lar12b] Kasper Green Larsen. Higher cell probe lower bounds for evaluating polynomials. In *Foundations of Computer Science (FOCS), 2012 IEEE 53rd Annual Symposium on*, pages 293–301. IEEE, 2012.

[Liu04] Ding Liu. A strong lower bound for approximate nearest neighbor searching. *Information Processing Letters*, 92(1):23–29, 2004.

[Mil94] Peter Bro Miltersen. Lower bounds for union-split-find related problems on random access machines. In *Proceedings of the twenty-sixth annual ACM symposium on Theory of computing*, pages 625–634. ACM, 1994.

[MNSW95] Peter Bro Miltersen, Noam Nisan, Shmuel Safra, and Avi Wigderson. On data structures and asymmetric communication complexity. In *Proceedings of the twenty-seventh annual ACM symposium on Theory of computing*, pages 103–111. ACM, 1995.

[PD06] Mihai Patrascu and Erik D Demaine. Logarithmic lower bounds in the cell-probe model. *SIAM Journal on Computing*, 35(4):932–963, 2006.

[PT06a] Mihai Pătraşcu and Mikkel Thorup. Time-space trade-offs for predecessor search. In *Proceedings of the thirty-eighth annual ACM symposium on Theory of computing*, pages 232–240. ACM, 2006.

[PT06b] Monica Patrascu and Mikkel Thorup. Higher lower bounds for near-neighbor and further rich problems. In *Foundations of Computer Science, 2006. FOCS'06. 47th Annual IEEE Symposium on*, pages 646–654. IEEE, 2006.

[PTW10] Rina Panigrahy, Kunal Talwar, and Udi Wieder. Lower bounds on near neighbor search via metric expansion. In *Foundations of Computer Science (FOCS), 2010 51st Annual IEEE Symposium on*, pages 805–814. IEEE, 2010.

[Yao81] Andrew Chi-Chih Yao. Should tables be sorted? *Journal of the ACM (JACM)*, 28(3):615–628, 1981.