

**The Bit Probe Model for Membership Queries:
Nonadaptive Bit Queries
Exposition by William Gasarch**

1 Introduction

This is an exposition of parts of the excellent papers [1],[2], and private email with Peter Bro Miltersen.

Def 1.1 If $x \in \mathbb{N}$ then $[x] = \{1, 2, \dots, x\}$.

Consider the following problem: The universe is $[U]$. You want to store a subset of $[U]$ of size $\leq n$ such that you do not use too much space, and the question “ $x \in A?$ ” can be answered easily.

Example 1.2 Let $A \subseteq [U]$ of size $\leq n$.

1. Store the $|A|$ elements in sorted order in the first $|A|$ cells of an array of length n . Store NULL in the other cells.
 - Probes needed to determine $e \in A$: $O(\log n)$. Note that the queries are adaptive.
 - Space needed to store: n cells of $O(\log U)$ bits each.
2. Use an array of U cells of 1-bit each. $U[a] = 1$ iff $a \in A$.
 - Probes needed to determine $e \in A$: 1.
 - Space needed to store: U cells of 1 bits each.

We will look at variants of bit vectors. Is there a way to still have $O(1)$ bit queries but use $\ll U$ cells of 1-bit each? We will see in Section 2 that the answer is YES.

We will now define the bit probe model. We only define and (for now) work with the non-adaptive case where all of the queries are made at the same time. Formally we should include the phrase “nonadaptive” but will omit it in this treatment.

Def 1.3 An $(U, n; s, q)$ *bit probe data structure for membership* (henceforth BPDS) consists of the following. (U is the size of the *universe*, n is the size of sets we will be storing, s is the number of bits in the data structure (s stands for *space*), and q is the number of bit *queries*.)

1. A function that will, given $A \subseteq [U]$ of size $\leq n$, output a vector $CELL \in \{0, 1\}^s$.
2. (Assume that the function in step 1 has been done.) Two functions:
 - (a) A function that takes as input $u \in [U]$ and outputs (i_1, \dots, i_q) where $1 \leq i_1, \dots, i_q \leq s$. The intuition is that on input u we ask the bit queries $CELL[i_1], \dots, CELL[i_q]$. When $q = 2$ we will use (a_u, b_u) .

- (b) A function that takes input $u \in [U]$ and outputs a boolean function f_u on q boolean variables. We require that

$$u \in U \text{ iff } f_u(CELL[i_1], \dots, CELL[i_q]) = IN.$$

(We use IN for TRUE and OUT for FALSE since the real info is IN and OUT.)

Note 1.4

1. The function and process in Definition 1.3 need not be computable. Hence our lower bounds will be very strong. Our upper bounds will use easily computable functions.
2. The process in Definition 1.3 was nonadaptive: questions asked could not depend on prior answers given. If the questions can depend on prior ones then this is called adaptive. We will not be studying this.
3. We will often use the notation s and q rather than $s(n, U)$ and $q(n, U)$.
4. Assume that we have a $(U, n; s, 2)$ BPDS. Assume that to make the membership query “ $2 \in A?$ ” we need to make the bit queries $CELL[12]$ and $CELL[84]$. The following is possible:

- (a) If the set $\{1, 2, 7\}$ is stored then $CELL[12] = 1$ and $CELL[84] = 0$.
- (b) If the set $\{2, 4, 9\}$ is stored the $CELL[12] = 0$ and $CELL[84] = 1$.

In this case it may be that the membership query algorithm on the question “ $2 \in A?$ ” might tell you to, after making the bit-probe queries $CELL[12]$ and $CELL[84]$, XOR the bits together to get the answer to the membership query. The point is that, given a number, you know which queries are asked, but how those bits are set may depend on which set was stored.

5. The following is possible:
 - (a) If the membership query “ $17 \in A?$ ” is made then the bit probe queries asked are $CELL[4]$ and $CELL[8]$.
 - (b) If the membership query “ $25 \in A?$ ” is made then the bit probe queries asked are $CELL[4]$ and $CELL[8]$.
 - (c) How could this be? Well, it could be that

$$17 \in A \text{ iff } CELL[4] = 1 \text{ xor } CELL[8] = 0$$

and

$$25 \in A \text{ iff } CELL[4] = 0$$

In this case here is how the types of sets are stored: (1) If $17, 25 \notin A$ then set $CELL[4] = 1$ and $CELL[8] = 1$. (2) If $17 \in A$ and $25 \notin A$ then set $CELL[4] = 1$ and $CELL[8] = 0$. (3) If $17 \notin A$ and $25 \in A$ then set $CELL[4] = 0$ and $CELL[8] = 0$. (4) If $17, 25 \in A$ then set $CELL[4] = 0$ and $CELL[8] = 1$. The point is that you can have two different membership queries use the same two bit probe queries.

Example 1.5

1. Example 1.2.1 did not fit the model since the content of the cells was elements of $[U]$ not elements of $\{0, 1\}$.
2. Example 1.2.2 used $s(n, U) = U$ and $q(n, U) = 1$.

In this exposition we prove the following.

1. There is a $(U, n; c_0 U^\delta, q)$ -BPDS where $\delta = \frac{1}{\lceil q/n \rceil}$.
2. If there is a $(U, n; s, q)$ -BPDS then $s \geq c_0 U^{1/q}$.
3. There is a probabilistic $(U, n; s, q)$ -BPDS FILL IN.
4. If there is a $(U, n; s, 2)$ -BPDS and $n \geq 4$ then $s \geq U$.
5. If there is a $(U, n; s, 3)$ -BPDS then $s \geq c_0 n^{2/3} U^{1/3}$.
6. If there is a $(U, n; s, 3)$ -BPDS and $n \geq 16 \log U$ then $s \geq c_0 \frac{n^{1/2} U^{1/2}}{(\log U)^{1/2}}$.
7. There is a $(U, n; s, 4)$ BPDS where $s = c_0 n^{1/6} U^{5/6}$.
8. There is a $(U, n; s, 4)$ BPDS where $s =$. FILL IN

2 Easy Upper Bound: There is a $(U, n; c_0 U^\delta, q)$ -BPDS where $\delta = \frac{1}{\lceil q/n \rceil}$

Lemma 2.1 *Assume there is a map from $[U]$ to $\binom{[s]}{q}$. We will denote the set u maps to by B_u . Assume that, for all $u, u_1, \dots, u_n \in [U]$,*

$$|B_u \cap \bigcup_{i=1}^n B_{u_i}| < q.$$

Then there is a $(U, n; s, q)$ BPDS.

Proof:*Setting up the Data Structure:*

Initially all of the cells are set to 0.

Let $A \subseteq [U]$, $|A| \leq n$.For each $u \in A$ set the bits of B_u to 1.*Making a Query*To ask " $u \in A$?" you make q probes to the bits specified by B_u . If all q are 1 then output YES else NO.*Why does this work?*Clearly if $u \in A$ then the answer returned will be YES.Let $A \subseteq \{u_1, \dots, u_n\}$. Let $u \notin \{u_1, \dots, u_n\}$. We need to show that if A is stored and " $u \in A$?" is asked then the answer will be NO. Note that

$$|B_u \cap \bigcup_{i=1}^n B_{u_i}| < q$$

Therefore the q bits of B_u cannot all be set to 1. ■**Theorem 2.2** *Let $n, q \in \mathbb{N}$. Let $\delta = \frac{1}{\lceil q/n \rceil}$. There exists constants U_0, c_0 such that, for all $U \geq U_0$, there is a $(U, n; c_0 U^\delta, q)$ BPDS.***Proof:**Let d be a quantity to be named later. Let $\delta = 1/(d+1)$. Let p be a prime such that $U^\delta \leq p \leq 2U^\delta$. Let *TUPLE* be an injection of $[U]$ to $[U^\delta] \times \dots \times [U^\delta]$ ($d+1$ times). If $TUPLE(u) = (a_d, \dots, a_0)$ then let

$$f_u(x) = a_d x^d + \dots + a_0.$$

Let

$$B_u = \{(1, p_u(1)), \dots, (q, p_u(q))\}.$$

Note that

$$|B_u \cap B_w| \leq d$$

Hence

$$|B_u \cap \bigcup_{i=1}^n B_{u_i}| \leq dn.$$

We need $dn < q$ in order to apply Lemma 2.1. Taking $d = \lceil q/n \rceil - 1$ will suffice. Note that $\delta = d+1 = \lceil q/n \rceil$. ■

3 Easy Lower Bounds: If there is a $(U, n; s, q)$ -BPDS then $s \geq cU^{1/q}$

Theorem 3.1 *Let $n, q \in \mathbb{N}$. Let M be the least number such that*

$$\binom{M}{0} + \cdots + \binom{M}{n} \geq 2^q + 1.$$

If there is an $(U, n; s, q)$ BPDS then $\binom{s}{q} \geq \frac{U}{M}$.

Proof:

Assume, by way of contradiction, that there is a $(U, n; s, q)$ BPDS with $\binom{s}{q} < \frac{U}{M}$

Let $f : [U] \rightarrow \binom{[s]}{q}$ that maps u to the set of queries that it asks. Since $\binom{s}{q} < \frac{U}{M}$ there exists M elements of $[U]$ that map to the same set of queries. Let the M elements of $[U]$ be $W = \{u_1, \dots, u_M\}$. Let the bit queries be $X = \{x_1, \dots, x_q\}$

Map

$$A \in \binom{[W]}{0} \cup \cdots \cup \binom{[W]}{n}$$

to the way the bits of X are set to store A . This function has a domain of size

$$\binom{M}{0} + \cdots + \binom{M}{n} \geq 2^q + 1$$

and a codomain of size 2^q . Hence there are two sets that map to the exact same structure. This causes a contradiction. ■

Corollary 3.2 *Let $n, q \in \mathbb{N}$, $n, q \leq 100$ (any constant would do). There exists U_0 and c_0 such that, for all $U \geq U_0$, if there is an $(U, n; s, q)$ BPDS then $s \geq c_0 U^{1/q}$.*

Proof:

Let M be the least number such that

$$\binom{M}{0} + \cdots + \binom{M}{n} \geq 2^q + 1.$$

Note that M is a constant. By Theorem 3.1 $\binom{s}{q} \geq \frac{U}{M}$. Note that $\binom{s}{q} \leq s^q$. Hence we have

$$s^q \geq \binom{s}{q} \geq \frac{U}{M}$$

$$s \geq \left(\frac{1}{M}\right)^{1/q} U^{1/q}$$

Let $c_0 = \left(\frac{1}{M}\right)^{1/q}$.

■

4 Probabilistic BPDS

5 Lower Bound: If there exists a $(U, n; s, 2)$ -BPDS and $n \geq 2$ then $s \geq U$

The proof of the following lemma is left to the reader.

Lemma 5.1 *If a graph has more edges than vertices then it must have a cycle.*

Theorem 5.2 *Let $n \geq 2$. In any $(U, n; s, 2)$ BPDS, $s \geq U$. (Note that there is no Ω in this result.)*

Proof:

We will prove this by induction on U . Note that $U = 1, n \geq 2$ makes sense since $A \subseteq U$ such that $|A| \leq n$. In that case $|A| \leq 1$.

Base Case: $U = 1$. If $s < U$ then $s = 0$. So no queries can be made. Thus if $A = \emptyset$ or $A = \{1\}$, the membership algorithm gives the same answer. This is a contradiction.

We do not need the $U = 2$ nor the $U = n$ case; however, we present it for enlightenment.

Case $U = 2$ For Fun!: $U = 2$. If $s < U$ then $s = 1$. Map $\emptyset, \{1\}$ and $\{2\}$ to how they are stored. Since there is only one bit, two of them map to the same storage. If its \emptyset and $\{1\}$ then the query “ $1 \in A?$ ” will be answered incorrectly. If its \emptyset and $\{2\}$ then the query “ $2 \in A?$ ” will be answered incorrectly. If its $\{1\}$ and $\{2\}$ then query “ $2 \in A?$ ” will be answered incorrectly.

Case $U = n$ For Fun!: (In this case we are storing all subsets of U .) If $s < U$ then we can take $s = n - 1$. Map all elements of 2^U to how the $n - 1$ bits are set. Since this map had domain of size 2^n and co-domain of size 2^{n-1} , two different sets map to the same setting. Call the sets A, B . Let $a \in A \oplus B$ (so either $a \in A - B$ or $a \in B - A$). The membership query algorithm will make a mistake on the query “ $a \in A?$ ”.

Induction Step: We can assume $U \geq 2$. Assume there exists a nonadaptive $(U, n; s, 2)$ bit-probe data structure where $s < U$.

Let $u \in [U]$. Note that to determine if $u \in U$ you make two queries, which we denote a_u and b_u (so the actual queries are to $CELL[a_u]$ and $CELL[b_u]$). Then you apply some boolean function of two variables to the answers, which we call $f_u(-, -)$. There are several case of what f_u can be which lead to several cases for our theorem. Most of them are easy.

1. There exists u such that f_u is the constant function. We take $f_u(x, y) = TRUE$ (FALSE is similar). Let $A_0 = \emptyset$. If you store A_0 in your data structure and ask “ $u \in A_0?$ ” you get answer TRUE. This is incorrect. Contradiction.
2. There exists u such that $f_u(x, y)$ depends on only one of the variables. We take $f_u(x, y) = x$ (the cases $f_u(x, y) = \neg x$, $f_u(x, y) = y$, and $f_u(x, y) = \neg y$ are similar). Hence we have

$$u \in [U] \text{ iff } a_u = 1.$$

Claim 1: If $A \subseteq [U] - \{u\}$ and $|A| \leq n$. When A is stored, $CELL[a_u] = 0$.

Proof of Claim 1:

Assume that when A is stored $CELL[a_u] = 1$. Then the query " $u \in A$?" will be answered YES when it should be NO.

End of Proof of Claim 1

We can now create a nonadaptive $(s-1, 2)$ bit-probe data structure where the universe is $[U] - \{u\}$. This will contradict the induction hypothesis. Use the same storage you did for U ; but do not use $CELL[a_u]$. If this cell is ever asked about then hardwire the answer 0 into it. Hence it does not count as a cell. Thus we use $s-1$ cells.

3. We are now in the case where all of the queries depend on both x and y . Create an edge-labelled multigraph with vertex set $[s]$ and edge set

$$E = \{(a_u, b_u) : u \in [U] \text{ this edge is labelled } u\}.$$

It is possible that two different elements of U ask the same two questions which is why it is a multi-graph. (If this puzzles you then see the last item in Note 1.4.)

KEY: This graph has s vertices but $U > s$ edges. Hence it must have a cycle. Let C be that cycle. By renaming let the cycle be $(x_1, x_2, x_3, \dots, x_L)$. We do not know what L is, nor do we care. Let the labels on the edges be u_1, \dots, u_L .

There are several cases.

- (a) The cycle contains an edge labelled u such that $f_u(x, y) = x \wedge y$. We assume $f_{u_1}(x, y) = x \wedge y$. The other cases are similar.

We build a set $A \subseteq [U]$, $|A| \leq n$, which will help us get a contradiction.

- i. Stage 1: $A = \{u_1\}$. $CELL[x_1] = 1$ and $CELL[x_2] = 1$ are forced.
- ii. Stage 2: If $CELL[x_2] = 1$ forces $u_2 \in A$ then we are done: the set $A = \{u_1\}$ will cause a mistake on the query " $u_2 \in A$?". Hence we can assume that $f_{u_2}(x, y)$ is not forced by the choice $x = 1$.

Claim 2: The decision to have $u_2 \notin A$ forces the value of $CELL[x_3]$.

Proof of Claim 2:

Assume not. Say that $A = \{u_1\}$ is stored. Clearly $CELL[x_1] = CELL[x_2] = 1$. What about $CELL[x_3]$. By assumption it is not forced. Hence both the data structures

$$CELL[x_1] = 1, CELL[x_2] = 1, CELL[x_3] = 1.$$

and

$$CELL[x_1] = 1, CELL[x_2] = 1, CELL[x_3] = 0.$$

correctly answer all membership queries for the set $\{u_1\}$.

How do we encode $B = \{u_1, u_2\}$? To get the data structure to answer yes to " $u_1 \in A$?", we must set $CELL[x_1] = CELL[x_2] = 1$. We know that the bit

probe queries made to answer “ $u_2 \in A$?” are $CELL[x_2]$ and $CELL[x_3]$. But whether you set $CELL[x_3]$ to 0 or 1 you will get the answer NO based on what we know about storing $\{u_1\}$. Hence $\{u_1, u_2\}$ cannot be stored.

End of Proof of Claim 2

Hence $CELL[x_3]$ is forced to some value.

- iii. For $i = 3$ to $L-1$ assume inductively that (1) $u_1 \in A$, (2) $u_2, u_3, \dots, u_{i-1} \notin A$, and (3) $CELL[x_1], CELL[x_2], \dots, CELL[x_i]$ have been forced.
- iv. Stage i . If the setting of $CELL[x_i]$ forces the status of u_i then we are done: one of the sets $\{u_1, u_i\}$ or $\{u_1\}$ will cause a mistake on the query “ $u_i \in A$?”. Let b be such that $CELL[x_i]$ be forced to be b . Do not put u_i into A . This forces $CELL[x_{i+1}]$ to some value (proof similar to claim proved above).

We now have that if $A = \{u_1\}$ then $CELL[x_1], \dots, CELL[x_L]$ are forced. This forces the status of $\{u_L\}$. Hence one of the sets $\{u_1\}$ or $\{u_1, u_L\}$ will cause a mistake on the query “ $u_L \in A$?”.

- (b) The cycle contains an edge labelled u such that $f_u(x, y)$ is one of the following: $x \wedge \neg y, \neg x \wedge y, \neg x \wedge \neg y, x \vee y, x \vee \neg y, \neg x \vee y, \neg x \vee \neg y$. These cases are just like case a. The key to case a is that there was a setting of $f_u(x, y)$ that forced both x and y . The same is true here.
- (c) For all edge labels u on the cycle f_u is of the form $x \oplus y$ or $\neg(x \oplus y)$. (These are the only that are not constant, depend on only one variable, or are covered in cases a and b.) Hence we have that there exists $b_1, \dots, b_L \in \{0, 1\}$ such that

$$\begin{aligned}
 A(u_1) &= x_1 + x_2 + b_1 \pmod{2} \\
 A(u_2) &= x_2 + x_3 + b_2 \pmod{2} \\
 A(u_3) &= x_3 + x_4 + b_3 \pmod{2} \\
 &\vdots \\
 A(u_L) &= x_L + x_1 + b_L \pmod{2}
 \end{aligned}$$

Note the following

- If $A = \emptyset$ then the bits must be set such that, for all $1 \leq i \leq L$, $x_i + x_{i+1} + b_i \equiv 0$. Summing over all $1 \leq i \leq L$ all of the x_i 's cancel and you get $\sum_{i=1}^L b_i \equiv 0$.
- If $A = \{u_1\}$ then $x_1 + x_2 + b_1 = 1$ but, for all $2 \leq i \leq L$, $x_i + x_{i+1} + b_i \equiv 0$. Summing over all $1 \leq i \leq L$ all of the x_i 's cancel and you get $\sum_{i=1}^L b_i \equiv 1$.

It cannot be that $\sum_{i=1}^L b_i$ is both $\equiv 0$ and $\equiv 1$ so this is a contradiction.

■

6 Lower Bounds: If there is a $(U, n; s, 3)$ -BPDS and $n \geq 4$ then $s \geq c_0 n^{2/3} U^{1/3}$

This section is based on [1].

We first prove a lemma that links lower bounds for nonadaptive $(U, n; s, q)$ BPDS to a problem in pure combinatorics.

Notation 6.1 A bipartite graph of the form $([U], [s], E)$ will be denoted (U, s, E) .

Def 6.2 A bipartite graph (U, s, E) where every element of U has degree q is called q -regular.

Def 6.3 Assume we have a $(U, n; s, q)$ BPDS. We associate to it the following bipartite graph: $G = (U, s, E)$ where

$$E = \{(u, x) : \text{to answer " } u \in A? \text{ " one of the bit-queries is } x \cdot \}$$

Note that every $u \in [U]$ has degree q . Also note that the bipartite graph has no information about how the answers to the queries are used. We refer to this graph as *the graph associated to the bit-probe data structure*.

Def 6.4 Let $G = (U, s, E)$.

1. If $u \in U$ then $N(u)$ is the set of all neighbors of u .
2. Let $Y \subseteq [s]$. Then

$$\text{ANSBY}(Y) = \{u \in U : N(u) \subseteq Y\}.$$

Def 6.5 Let $G = (U, s, E)$ and $n \in \mathbb{N}$. G is n -nice if, for all $Y \in \binom{[s]}{\leq n-1}$, $|\text{ANSBY}(Y)| \leq |Y|$.

Lemma 6.6 Assume there exists an $(U, n; s, q)$ BPDS. Then there is a q -regular bipartite graph $G = (U, s, E)$ that is n -nice.

Proof: Assume there exists an $(U, n; s, q)$ BPDS. Let $G = (U, s, E)$ be the associated bipartite graph.

Assume, by way of contradiction, that there exists $Y \in \binom{[s]}{\leq n-1}$, such that $|\text{ANSBY}(Y)| \geq |Y| + 1$. Let Z be a subset of $\text{ANSBY}(Y)$ of size $|Y| + 1$. Note that $|Z| \leq n$. Hence every $A \subseteq Z$ has a representation in the data structure. Note also that the answers given by the data structure for the elements of Z depend only on the bits of Y .

Let $A \subseteq Z$. If A is the set you are trying to store then you will set the bits of Y . Map each such A to the way you set the bits of Y . This is a mapping of a set of size $2^{|Y|+1}$ to a set of size $2^{|Y|}$. Hence two different $A_1, A_2 \subseteq [Z]$ set the bits the exact same way. This will cause some membership query to be answered incorrectly, and is hence a contradiction. ■

We will now prove a combinatorial lemma which will enable us to get a lower bound.

6.1 The Lower Bounds

Lemma 6.7 *Let $s, n, U \in \mathbb{N}$. Let $4 \leq n \leq U$. If $G = (U, s, E)$ is any n -nice 3-regular bipartite graph then $s \geq n^{2/3}U^{1/3}/4$. (Note that there is no Ω in this result.)*

Proof:

Let $G = (U, s, E)$ be a 3-regular n -nice bipartite graph. We will pick a set $Y \subseteq [s]$, $|Y| = n - 1$, at random and find the expected value of ANSBY(Y).

Fix $u \in U$. Note that $|N(u)| = 3$.

$$\begin{aligned} \Pr(N(u) \subseteq Y) &= \frac{\binom{s-3}{n-4}}{\binom{s}{n-1}} = \frac{(s-3)!}{(n-4)!(s-n+1)!} \frac{(n-1)!(s-n+1)!}{s!} = \\ &= \frac{(n-1)(n-2)(n-3)}{s(s-1)(s-2)} \geq \left(\frac{n-3}{s}\right)^3. \end{aligned}$$

Hence

$$\Pr(N(u) \subseteq Y) \geq \left(\frac{n-3}{s}\right)^3.$$

Let EX be the expected number of u such that $N(u) \subseteq Y$. By the above calculation

$$EX \geq U \left(\frac{n-3}{s}\right)^3.$$

Since G is n -nice we know that $EX \leq n - 1$. Hence

$$U \left(\frac{n-3}{s}\right)^3 \leq n - 1.$$

$$U^{1/3} \frac{n-3}{s} \leq (n-1)^{1/3}.$$

$$\frac{n-3}{s} \leq \frac{(n-1)^{1/3}}{U^{1/3}}.$$

$$\frac{s}{n-3} \geq \frac{U^{1/3}}{(n-1)^{1/3}}.$$

$$s \geq \frac{U^{1/3}(n-3)}{(n-1)^{1/3}}.$$

$$s \geq \frac{U^{1/3}(n-3)}{(n-1)^{1/3}}.$$

Since $n \geq 4$ we have $n - 3 \geq \frac{n}{4}$. Clearly $n - 1 \leq n$. Hence we have

$$s \geq \frac{U^{1/3}n}{4n^{1/3}} \geq n^{2/3}U^{1/3}/4$$

■

Theorem 6.8 *If there is a $(U, n; s, 3)$ BPDS and $n \geq 4$, then $s \geq n^{2/3}U^{1/3}/4$. (Note that there is no Ω in this result.)*

Proof: This follows from Lemma 6.6 and Lemma 6.7. ■

7 Lower Bounds: If there is a $(U, n; s, 3)$ -BPDS and $n \geq 16 \log U$ then $s \geq \Omega\left(\frac{n^{1/2}U^{1/2}}{(\log U)^{1/2}}\right)$

FILL IN

Theorem 7.1 *There exists U_0 and c_0 such that, for all $U \geq U_0$, if there is an $(U, n; s, 3)$ BPDS with $n \geq 16 \log U$, then*

$$s \geq c_0 \frac{n^{1/2}U^{1/2}}{(\log U)^{1/2}}.$$

8 There is a $(U, n; s, 4)$ -BPDS where $s =$

8.1 Combinatorial Set Up

Def 8.1 Let $G = (U, s, E)$ be a bipartite graph. Let $A \subseteq U$. Then G_A is the labeled bipartite graph where the elements of A are labeled IN and the elements in $[U] - A$ are labeled OUT.

Def 8.2 Let $a, b, q \in \mathbb{N}$ such that $0 < a, b < q$. Let $G = (U, s, E)$ be a q -regular labeled bipartite graph. Let $A \subseteq U$. An (a, b) -coloring of G_A is a partial 2-coloring of $[s]$, using the colors $\{0, 1\}$ such that the following occurs.

1. Every $u \in A$ then at least a neighbors of u are colored 1.
2. Every $u \notin A$ then at least b neighbors of u are colored 0.

Note 8.3 We allow the 2-coloring of $[s]$ to be partial since all we care about is the conclusion that IF $u \in A$ then ... and IF $u \notin A$ then

Def 8.4 Let $a, b, q \in \mathbb{N}$ such that $0 < a, b < q$. Let $G = (U, s, E)$.

1. G is (a, b, q) -useful if it is q -regular and has the following property: for any subset $A \subseteq [U]$, G_A is (a, b) -colorable.
2. G is $(n; a, b, q)$ -useful if it is q -regular and has the following property: for any subset $A \subseteq [U]$, $|A| \leq n$, G_A is (a, b) -colorable.

Lemma 8.5 *If there exists a 4-regular $(n; 3, 2, 4)$ -useful bipartite graph $G = (U, s, E)$ then there is a $(U, n; s, 4)$ BPDS.*

Proof:

Setup: Let $A \subseteq U$, $|A| \leq n$. Let COL be the $(3, 2)$ -coloring of G_A . For each $x \in [s]$ that is colored 0 (1) let the corresponding bit be 0 (1).

Query: To determine if $u \in A$ ask the 4 bit-queries that correspond to the elements in $N(u)$. If at least 3 of them are 1 then answer IN. If at most 2 of them are 1 then answer OUT.

Why does this work?

If $u \in A$ then A was labeled IN. Hence at least 3 of its neighbors are colored 1. Hence the algorithm will return IN.

If $u \notin A$ then A was labeled OUT. Hence at least 2 of its neighbors are colored 0. Hence at most 2 neighbors are colored 1. Hence the algorithm will return OUT. ■

Note 8.6 Lemma 8.5 can be generalized.

So our job is reduced to finding such bipartite graph.

8.2 $\frac{14^+}{5}$ -Expansion Implies $(3, 2, 4)$ -Useful

Def 8.7 Let $\alpha > 1$. Let $G = (U, s, E)$ be a bipartite graph.

1. G has *expansion* α if for every $A \subseteq [U]$, $|N(A)| \geq \alpha|A|$.
2. G has *expansion* α^+ if for every $A \subseteq [U]$, $|N(A)| > \alpha|A|$.

Note 8.8 Looking at the definition of expansion it looks like you would need $s \geq U$ to have an α -expanding graph. This is true. Hence it would seem that such graphs are not useful to us. But they will be later as a subroutine of what we need to do.

Lemma 8.9 *Let $G = (U, s, E)$ be a 4-regular bipartite graph with expansion $\frac{14^+}{5}$.*

1. G is $(3, 2, 4)$ -useful.

2. Let $A \subseteq [U]$. Informally, we want to say that a partial $(3, 2)$ -coloring of G_A can be extended to a full $(3, 2)$ -coloring of G_A . We proceed formally. Let $A \subseteq U$. Assume the right hand side of G has been partially $(3, 2)$ -colored such that

- (a) if $u \in U$ is IN then either at least 3 elements of $N(u)$ are colored 1 or no elements of $N(u)$ are colored 0, and
- (b) if $u \in U$ is OUT then either at least 2 elements of $N(u)$ are colored 0 or no elements of $N(u)$ are colored 1.

Then this partial coloring can be extended to a full $(3, 2)$ coloring.

Proof:

We prove part (2). Part (1) follows. Let $A \subseteq [U]$. Assume there already is a partial coloring as described in the premise. The following algorithm will complete the $(3, 2)$ -coloring.

1. Initialize *ACTIVE* to be the union of the following two sets.

$$\{u \in U : u \text{ is labeled IN and } \leq 2 \text{ of its neighbors are colored 1 } \}$$

$$\{u \in U : u \text{ is labeled OUT and } \leq 1 \text{ of its neighbors are colored 0 } \}$$

2. Initialize *NCOL* to be

$$NCOL = \{x \in [s] : x \text{ has not been colored yet } \}.$$

3. For all $x \in NCOL$

- (a) If $N(x) \cap ACTIVE$ are all IN then $COL(x) = 1$ and $NCOL = NCOL - \{x\}$.
- (b) If $N(x) \cap ACTIVE$ are all OUT then $COL(x) = 0$ and $NCOL = NCOL - \{x\}$.

4. For all $u \in ACTIVE$

- (a) If at least 3 elements of $N(u)$ are colored 1 then $ACTIVE = ACTIVE - \{u\}$.
- (b) If at least 2 elements of $N(u)$ are colored 0 then $ACTIVE = ACTIVE - \{u\}$.

5. If $ACTIVE \neq \emptyset$ then GOTO Step 3

Clearly all $u \notin ACTIVE$ that are labeled IN (OUT) have at least 3 (2) neighbors colored 1 (0). We need to show that eventually $ACTIVE = \emptyset$. (Note that this is all we need- we do not need that $NCOL = \emptyset$ since the coloring can be partial.) We show that so long as $ACTIVE \neq \emptyset$ either some $u \in [U]$ will become inactive or some $x \in [s]$ will get colored.

Assume $ACTIVE \neq \emptyset$ but during an iteration of the algorithm no element of $[s]$ is colored and no element of $[U]$ is made inactive. We will get a contradiction. Let $ACTIVE = ACT_{IN} \cup ACT_{OUT}$ where the IN (OUT) elements of $ACTIVE$ are ACT_{IN} (ACT_{OUT}).

We count $|N(ACT_{IN} \cup ACT_{OUT})|$ in two different ways.

1. Every elements of ACT_{IN} has 4 neighbors. Look at $u \in ACT_{OUT}$. We look at the neighbors of u and determine how many of them are not already counted in $N(ACT_{IN})$. No neighbor of u is colored 1 since $u \in ACT_{OUT}$. At most one neighbor of u is colored 0 (if two were colored 0 then $u \notin ACTIVE$). Of the non-colored neighbors of u all of them are in $N(ACT_{IN})$, else they would have been colored. Hence

$$|N(ACT_{IN} \cup ACT_{OUT})| \leq 4|ACT_{IN}| + |ACT_{OUT}|.$$

Hence (we'll need this later)

$$2|N(ACT_{IN} \cup ACT_{OUT})| \leq 8|ACT_{IN}| + 2|ACT_{OUT}|.$$

2. Every elements of ACT_{OUT} has at most 4 neighbors. Look at $u \in ACT_{IN}$. We look at the neighbors of u and determine how many of them are not already counted in $N(ACT_{OUT})$. No neighbor of u is colored 0 since $u \in ACT_{IN}$. At most two neighbor of u are colored 1 (if three were colored 1 then $u \notin ACTIVE$). Of the non-colored neighbors of u all of them are in $N(ACT_{OUT})$, else they would have been colored. Hence

$$|N(ACT_{IN} \cup ACT_{OUT})| \leq 2|ACT_{IN}| + 4|ACT_{OUT}|.$$

Hence (we'll need this later)

$$3|N(ACT_{IN} \cup ACT_{OUT})| \leq 6|ACT_{IN}| + 12|ACT_{OUT}|.$$

Adding together the two equations that we claimed we would need later you get

$$5|N(ACT_{IN} \cup ACT_{OUT})| \leq 14|ACT_{IN}| + 14|ACT_{OUT}|.$$

or

$$|N(ACT_{IN} \cup ACT_{OUT})| \leq \frac{14}{5}(|ACT_{IN}| + |ACT_{OUT}|).$$

Hence if $X = ACT_{IN} \cup ACT_{OUT}$ then $|N(X)| \leq \frac{14}{5}|X|$. This contradicts G being $\frac{14^+}{5}$ -expanding. ■

This looks good: if we can find a graph $G = (U, s, E)$ that is 4-regular and $\frac{14^+}{5}$ -expanding then it will be $(3, 2, 4)$ -useful. However, since the expansion condition as we have stated it applies even to the entire set U , the resulting data structure would use space greater than $\frac{14U}{5}$, which is no good. But we only need an $(n; 3, 2, 4)$ -useful graph, and for that, a weaker condition will suffice.

8.3 $(2n, \frac{14^+}{5})$ -Expansion implies $(n; 3, 2, 4)$ -Useful

Def 8.10 Let $\alpha > 1$ and $L \in \mathbb{N}$.

1. $G = (U, s, E)$ has *expansion* (L, α) if for every $A \subseteq [U]$, $|A| \leq L$, $|N(A)| \geq \alpha|A|$.
2. $G = (U, s, E)$ has *expansion* (L, α^+) if for every $A \subseteq [U]$, $|A| \leq L$, $|N(A)| > \alpha|A|$.

Lemma 8.11 Let $G = (U, s, E)$ be a 4-regular bipartite graph with expansion $(2n, \frac{14^+}{5})$. Then G is $(n; 3, 2, 4)$ -useful.

Proof:

Let $A \subseteq [U]$ such that $|A| \leq n$. We exhibit a $(3, 2)$ coloring of G_A . Actually we just do the algorithm from Lemma 8.9. We need to show that eventually $ACTIVE = \emptyset$.

Assume $ACTIVE \neq \emptyset$ but during an interaction of the algorithm no element of $[s]$ is colored and no element of $[U]$ is made inactive. We will get a contradiction. Let $ACTIVE = ACT_{IN} \cup ACT_{OUT}$ where the IN (OUT) elements of $ACTIVE$ are ACT_{IN} (ACT_{OUT}).

Claim 1: $|ACT_{OUT}| < |ACT_{IN}|$.

Proof of Claim 1:

Assume, by way of contradiction, that $|ACT_{OUT}| \geq |ACT_{IN}|$.

Choose $V \subseteq ACT_{OUT}$ of exactly the same size as ACT_{IN} . Note that $|V| = |ACT_{IN}| \leq n$ (since originally $|A| \leq n$).

so

$$|ACT_{IN} \cup V| \leq 2n$$

Hence, since G is $(2n, \frac{14^+}{5})$ -expanding,

$$|N(ACT_{IN} \cup V)| > \frac{14}{5}(|ACT_{IN}| + |V|).$$

Since $|V| = |ACT_{IN}|$ we have

$$|N(ACT_{IN} \cup V)| > \frac{28}{5}(|ACT_{IN}|).$$

This is what we will contradict.

By the same reasoning used in Lemma 8.9 we have that

$$|N(ACT_{IN} \cup V)| \leq 4|ACT_{IN}| + |V| = 5|ACT_{IN}|.$$

Note that we have

$$\frac{28}{5}|ACT_{IN}| < |N(ACT_{IN} \cup V)| \leq 5|ACT_{IN}|.$$

This is a contradiction.

End of Proof of Claim 1

The set of active elements of $[U]$ has size $\leq 2n$. Hence the bipartite graph that is left to color has $\leq 2n$ elements in the left hand side. By the premise this graph is α -expanding. It is partially colored. By Lemma 8.9 it can be extended to a full coloring. Formally this contradicts the assumption that the coloring stopped. ■

Note 8.12 Lemma 8.11 used the $\alpha > \frac{14}{5}$. This is not optimal.

8.4 There exists a Bipartite Graph Such That...

Now we need to show that such a graph exists.

Lemma 8.13 *There exists a constant c, U_0 such that the following is true. For every $\delta > 0$, for every U, n with $U \geq n$, and $U \geq U_0$, for every $s \geq cn^{\delta/(1+\delta)}U^{1/(1+\delta)}$ there is a 4-regular $(2n, (3 - \delta)^+)$ -expanding bipartite graph $G = (U, s, E)$.*

Proof:

We show the graph exists by the probabilistic method. For every $u \in [U]$ pick 4 elements of $[s]$ at random to be its neighbors. Let $\Delta = (3 - \delta)$.

We bound the expected number of X such that $|X| \leq 2n$ and $|N(X)| \leq \Delta|X|$. Since $|N(X)|$ is an integer this is equivalent to $|N(X)| \leq \lfloor \Delta|X| \rfloor$. Let $1 \leq r \leq 2n$. We first bound the expected number of X , $|X| = r$, such that $|N(X)| < \lfloor \Delta r \rfloor$.

There are $\binom{s}{\lfloor \Delta r \rfloor}$ subsets of $[s]$ of size $\lfloor \Delta r \rfloor$. Let Y be one of them. What is the probability that $N(X) \subseteq Y$? It is

$$\left(\frac{\binom{\lfloor \Delta r \rfloor}{4}}{\binom{s}{4}} \right)^r = \left(\frac{\lfloor \Delta r \rfloor (\lfloor \Delta r \rfloor - 1)(\lfloor \Delta r \rfloor - 2)(\lfloor \Delta r \rfloor - 3)}{s(s-1)(s-2)(s-3)} \right)^r$$

There exists a constant c_1 such that

$$\left(\frac{\lfloor \Delta r \rfloor (\lfloor \Delta r \rfloor - 1)(\lfloor \Delta r \rfloor - 2)(\lfloor \Delta r \rfloor - 3)}{s(s-1)(s-2)(s-3)} \right)^r \leq c_1 \left(\frac{\lfloor \Delta r \rfloor}{s} \right)^{4r}.$$

Hence the expected number of such X (of size r) is bounded above by

$$c_1 \binom{s}{\lfloor \Delta r \rfloor} \left(\frac{\lfloor \Delta r \rfloor}{s} \right)^{4r}$$

Hence the total number of such X (of size $1 \leq r \leq 2n$) is

$$c_1 \sum_{r=1}^{2n} \binom{U}{r} \binom{s}{\lfloor \Delta r \rfloor} \left(\frac{\lfloor \Delta r \rfloor}{s} \right)^{4r}.$$

By Stirling's formula there exists constants c_2, c_3

$$\binom{U}{r} \leq c_2 \left(\frac{eU}{r} \right)^r$$

$$\binom{s}{\lfloor \Delta r \rfloor} \leq c_3 \left(\frac{es}{\lfloor \Delta r \rfloor} \right)^{\lfloor \Delta r \rfloor}$$

(The e really is the e you know: 2.718...)

Putting this all together, and letting $c_4 = c_1 c_2 c_3$, we get that the expected number of X is bounded above by

$$c_4 \sum_{r=1}^{2n} \left(\frac{eU}{r} \left(\frac{es}{\lfloor \Delta r \rfloor} \right)^{\Delta} \left(\frac{\lfloor \Delta r \rfloor}{s} \right)^4 \right)^r$$

We look at the summand.

$$\left(\frac{eU}{r} \left(\frac{es}{\lfloor \Delta r \rfloor} \right)^{\Delta} \left(\frac{\lfloor \Delta r \rfloor}{s} \right)^4 \right)^r \leq \left(\frac{e^{\Delta+1} U (\lfloor \Delta r \rfloor)^{4-\Delta}}{r s^{4-\Delta}} \right)^r \leq \left(\frac{e^{\Delta+1} \Delta^{4-\Delta} U r^{3-\Delta}}{s^{4-\Delta}} \right)^r.$$

Since $\Delta = 3 - \delta$ and $r \leq 2n$ we get that this quantity is bounded above by

$$\left(\frac{e^{4-\delta} (3-\delta)^{1+\delta} U r^\delta}{s^{1+\delta}} \right)^r \leq \left(\frac{e^{4-\delta} (3-\delta)^{1+\delta} U (2n)^\delta}{s^{1+\delta}} \right)^r = \left(\frac{e^{4-\delta} (3-\delta)^{1+\delta} 2^\delta U (n)^\delta}{s^{1+\delta}} \right)^r$$

Let c_5 be a bound on $e^{4-\delta} (3-\delta)^{1+\delta} 2^\delta$ that is independent of n, U and even δ . Then the quantity above is bounded above by

$$\left(\frac{c_5 U n^\delta}{s^{1+\delta}} \right)^r$$

Hence we have that the expected number of such X is bounded above by

$$c_4 \sum_{r=1}^{2n} \left(\frac{c_5 U n^\delta}{s^{1+\delta}} \right)^r.$$

The sum is geometric. We (cleverly!) set the multiplier to be $\frac{1}{2c_4}$ and then sum the series.

$$\begin{aligned} \frac{c_5 U n^\delta}{s^{1+\delta}} &\leq \frac{1}{2c_4} \\ 2c_4 c_5 U n^\delta &\leq s^{1+\delta} \\ (2c_4 c_5 U n^\delta)^{1/(1+\delta)} &\leq s \end{aligned}$$

Let $c_6 = 2c_4 c_5$. We have that if $s \geq c_6 n^{\delta/(1+\delta)} U^{1/(1+\delta)}$ there is a 4-regular $(2n, 3 - \delta)$ -expanding bipartite graph. ■

Theorem 8.14 *There exists U_0 and c_0 such that, for all $U \geq U_0$, for all n , there is a $(U, n; s, 4)$ BPDS where $s = \lceil c_0 n^{1/6} U^{5/6} \rceil$.*

Proof: This follows from Lemmas 8.5, 8.11, and 8.13 with $\delta = 1/5$. ■

9 OTHER RESULT ON 4 probes

References

- [1] N. Alon and U. Feige. On the power of two, three, and four probes. In *Seventeenth Symposium on Discrete Algorithms: Proceedings of SODA '09*, 2009.
- [2] H. Huhman, P. Miltersen, J. Radhakrishnan, and S. Venkatesh. Are bit vectors optimal. *SIAM Journal on Computing*, 31:1723–1744, 2002. <http://www.daimi.au.dk/~bromille/Papers/index.html>.