Set membership with two classical and quantum probes

Speaker: Jaikumar Radhakrishnan
ICTS Bengaluru
TIFR Mumbai

Joint work with ..., ..., ..., ...,

Shyam Dhamapurkar

and

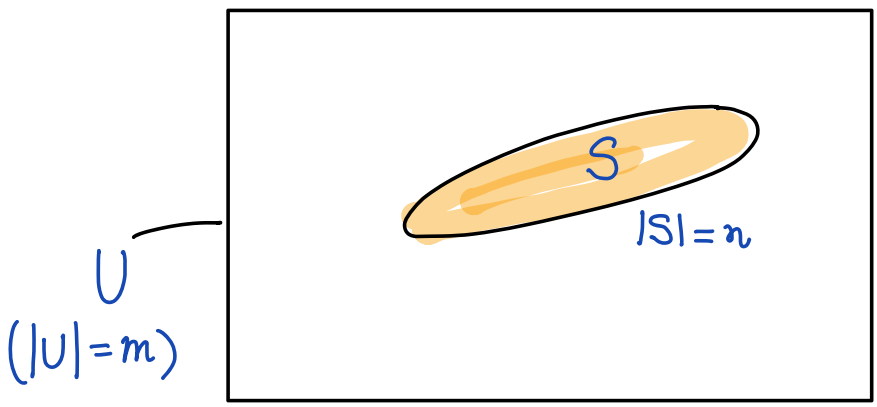
Shubham Pawar

IISc, 28 Oct 2022

The Problem

m: size of the universe

n: size of the subset



Assume $n << m$.

Task: Represent S as a bit string b.
Then, given $x \in [m]$, determine if $x \in S$, by reading at most t bits of b.

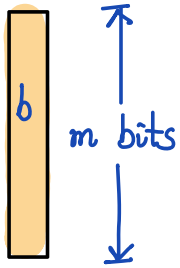
Goal: Study the trade-off between $|b|$ and t.

Examples

- **Characteristic vector**

  Can answer queries with just one probe.

  (n plays no role.)

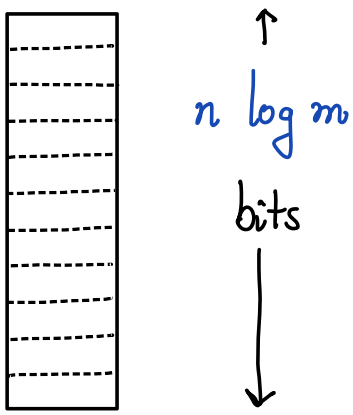


- **Sorted array**

  Can answer queries with

  $(\log n) \times (\log m)$

  bit probes.



Space, #probes

$S(m, n, t)$ = minimum number of bits of storage in a scheme that can answer membership queries with t bit probes.

universe size, set size

- Characteristic vector:

$$S(m, n, t=1) \leq m \qquad \forall m, n$$

- Sorted table:

$$S(m, n, t=(\log m)\cdot(\log n)) = O(n \log m \log n)$$

Question: Lower bounds? ... Upper bounds?

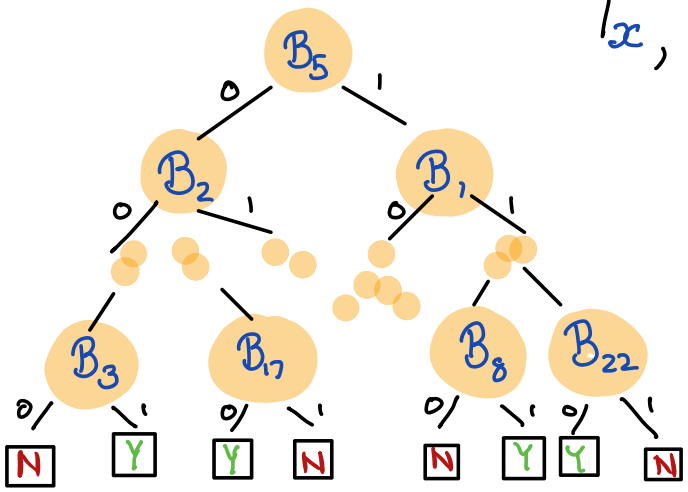
Lower Bounds

Theorem: $S(m, n, t) \geq t\, m^{1/t}\, n^{1-1/t}$

Proof: Fix a scheme that performs the task using s bits of storage.
We wish to show that s is large.

For each query of the form "Is x in S?", there is a query tree.

T_x, depth $= t$



We will use the polynomial method.

Idea: Associate with each set S a polynomial of degree at most nt and show that these polynomials are linearly independent. So,

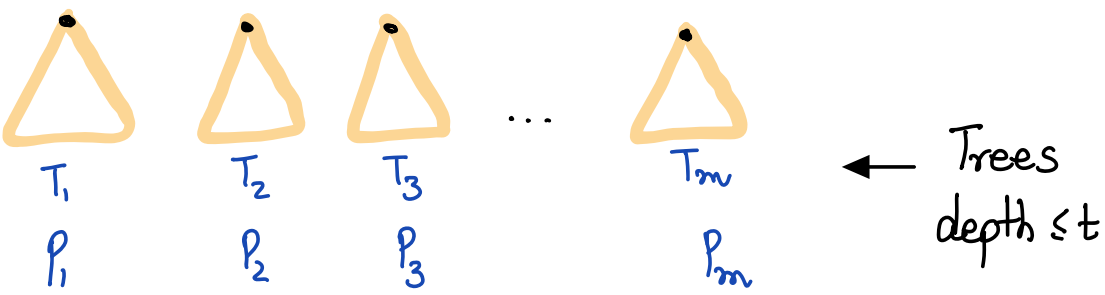
$$\binom{m}{n} \leq \binom{s+nt}{nt} \Rightarrow s \geq t\, m^{1/t}\, n^{1-1/t}$$

- Every path corresponds to a monomial of the form

$$B_5 (1-B_1) \cdots B_{52}$$

which evaluates to 1 iff the computation on the tree takes that path.

- For each $x \in [m]$, there is a polynomial $P_x(B_1, \ldots, B_s)$ such that P_x evaluates to 1 on $(y_1, y_2, \ldots, y_s) \in \{0,1\}^s$ iff T_x returns Y when the memory is assigned $(y_1, y_2, \ldots, y_s)$.



$$P_S(B_1, B_2, \ldots, B_s) = \prod_{i \in S} P_i(B_1, B_2, \ldots, B_s)$$

For each set S, we have a polynomial $p_S(B_1, \ldots, B_s)$ of degree at most nt.

CLAIM: These polynomials are linearly independent.

$$\sum_{\substack{S \\ |S| = n}} \alpha_S p_S(B_1, B_2, \ldots, B_s) \equiv 0 \quad \Rightarrow \quad \forall S \;\; \alpha_S = 0$$

Substitute for $B_1, \ldots, B_s$ the contents of the memory when S is stored.

So, $$\binom{m}{n} \leq \underbrace{\binom{s}{0} + \binom{s}{1} + \cdots + \binom{s}{nt}}_{\text{Total number of monomials.}} \leq \binom{s+nt}{nt}.$$

For t a constant and $n \leq m$,

$$s(m, n, t) \gtrsim t\, m^{1/t}\, n^{1-1/t}.$$

Note that for $t = 1$, no compression is possible.

How good is this lower bound for larger t?

Upper Bounds

Mohit Garg (PhD thesis, 2016)

$$3 \le t \le \frac{1}{10} \log\log n \qquad \text{(odd)}$$

$$n \le m^{1-\varepsilon}$$

$$s(m, n, t) = O\left(m^{\frac{2}{t+1}} n^{1-\frac{2}{t+1}} (\log m)^2\right)$$

What about $t = 2$?

$t = 2$, non-adaptive $\qquad s(m, n, t=2 \text{ non-adaptive}) = m \qquad (n \ge 2)$

$t = 2$, adaptive

Alon and Feige (2009)

$$s(m, n, 2) = O\left(mn \frac{\log\left(\frac{\log m}{n}\right)}{\log m}\right)$$

So, $s(m, n, 2) = o(m)$ whenever $n = o(\log m)$.

No savings when $n > \log m$!

(Mohit Garg's thesis)

$$m^{1-1/\lfloor n/4 \rfloor} \leq S(m, n, 2) \leq m^{1-1/4n+1}$$

No savings are possible if $n \geq \log m$

Better savings for small values of n.

Several works have focussed on improving the upper bound for specific values of n.

$S(m, n=3, t=2)$	$= \Theta(m^{2/3})$	[Baig+Kesh 2018, Kesh 2018]
$S(m, n=4, t=2)$	$= O(m^{5/6})$	[, Baig+Kesh+Sadani 2019]
$S(m, n=5, t=2)$	$= O(m^{5/6})$	[ ]

Recent (with Shyam Dhamapurkar and Shubham Pawar ICALP 2022)

Upper bound

$$S(m, n, t=2) = m^{1-1/n} \quad \text{for} \quad n = \pm 1 \pmod 3$$
$$= m^{1-1/(n-1)} \quad \text{for} \quad n = 0 \pmod 3$$

$(S(m, n = 8, 9, t=2) = O(m^{6/7}))$

Lower bound

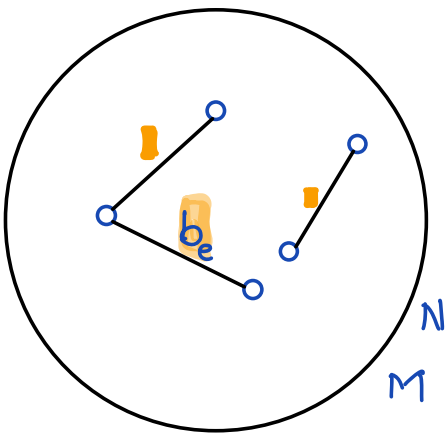
$$S(m, n, t=2) = \Omega\left(m^{1-2/(n+3)}\right).$$

Upper bound

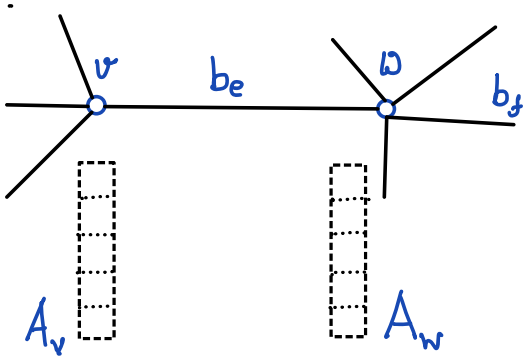
Based on dense graphs of high girth.

The idea

- Several elements x of the universe are assigned to the same edge e. They all read the bit b_e in their first probe.



- The bit b_e indicates where the answer to the query is stored.



- m/M elements read the same bit b_e.
- Each array A_v has m/M bits.

Total space = $M + N(m/M)$

Key lemma: Suppose H is a graph with M edges, of which n are colored GREEN and $M-n$ are colored RED. If $n \leq \lfloor 3g/4 \rfloor$, then the edges of H can be assigned directions such that if a GREEN edge points to a vertex v, then no other edge points to v.

The bound is tight.

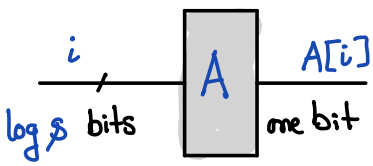
Annoying to prove!

If there are graphs with L vertices and $L^{1+\tau(g)}$ edges with girth $\geq g$, then

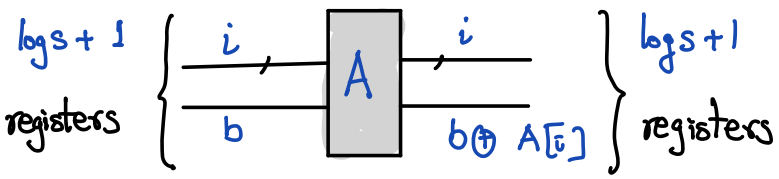
$$S(m, n, t=2) = O\left(m^{(1+\tau(4/3))/(1+2\tau(4/3))}\right)$$

Quantum schemes

- As before we represent the set as a bit string. A bit array with s bits is made available as the following reversible "quantum hardware".



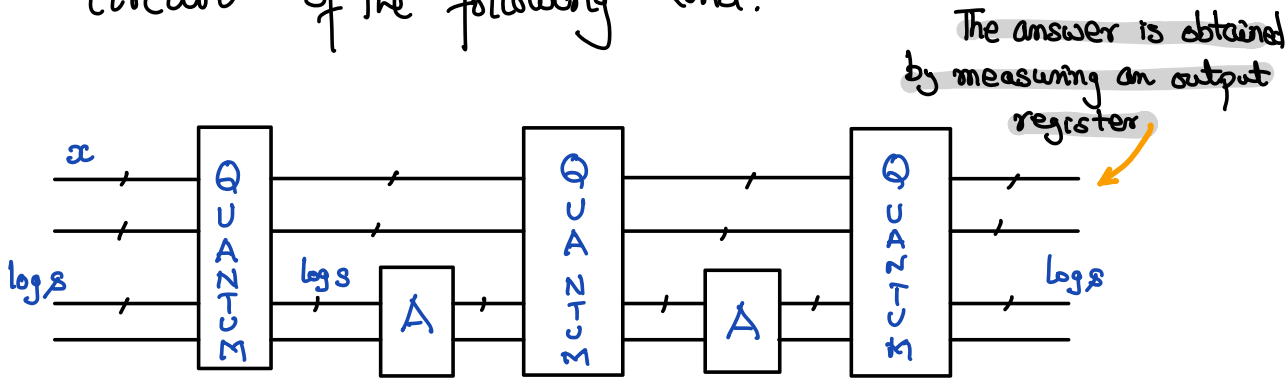
Classical



quantum

can be probed in superposition

- A two-probe quantum scheme corresponds to a circuit of the following kind.



quantum representation of S

A two-probe quantum scheme

$S^{Q}(m, n, t)$ = min. size of the storage A so that all queries are answered correctly (with probability 1).

$$S^{Q}(m, n, t) = \Omega(m^{1/t} n^{1-1/t})$$

Today: The power of two quantum probes.

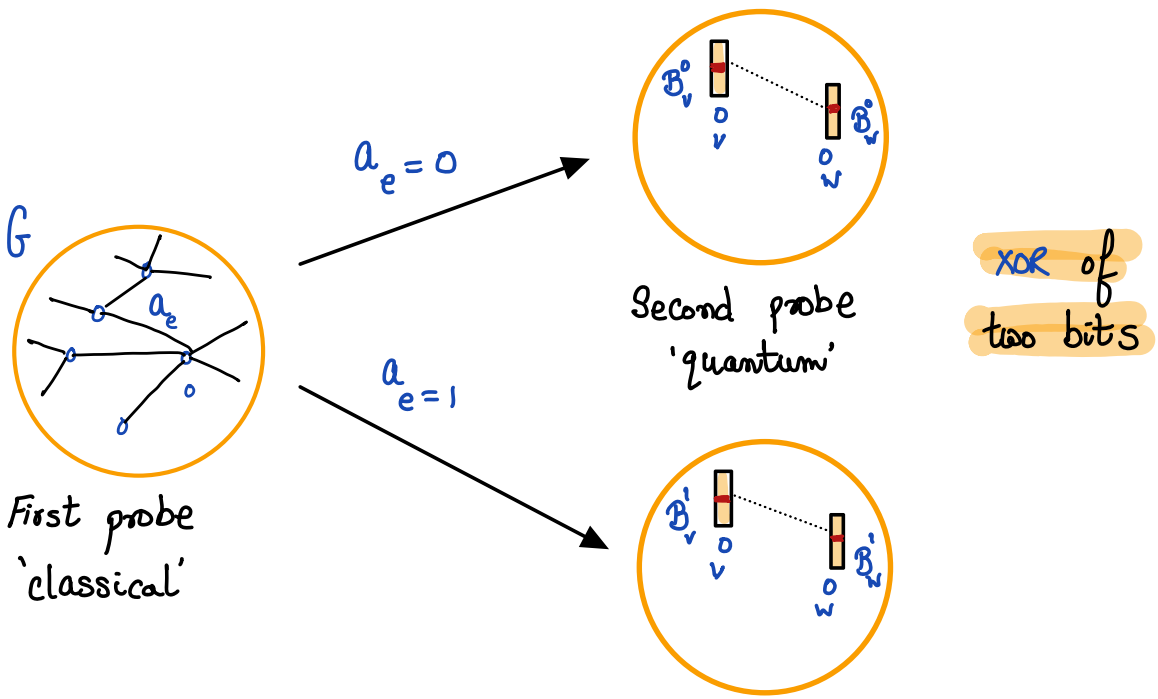
> Fact: Quantum algorithms can compute the XOR of two bits $A[i]$ and $A[j]$ by making just one probe into A.

Result

$$S^{Q}(m, n \le m^{1/8}, t=2) = O(m^{7/8})$$

- even for large sets
- only the second probe needs XOR
- significantly sublinear

IDEA: Similar to the classical scheme



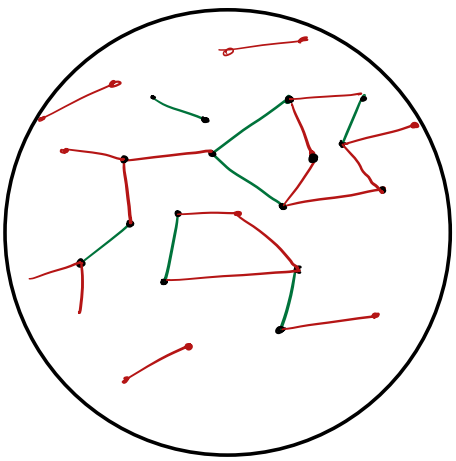
Space used

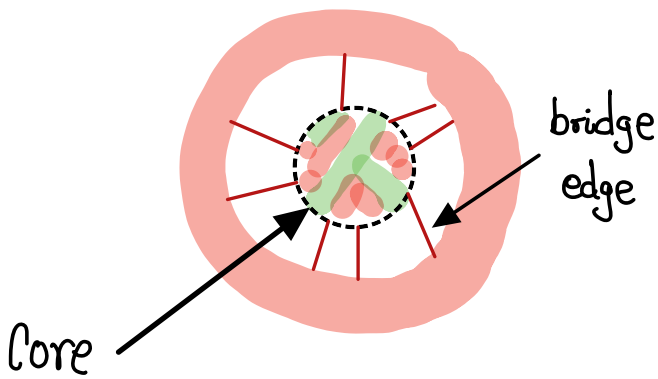
- Suppose G has N vertices and M edges.
- Total space = $M + 2N\left(\frac{m}{M}\right)$

Given a set S, how does assign set the bits (a_e) and (B_v^0, B_v') so that all queries are answered correctly?

Idea: $|S| = n$

- Call the edges where the elements in S make their first probe GREEN.
  Call the other edges RED.





- The GREEN edges can be gathered into a sparse core.
- Split the core into two forests.
- Add the bridge edges to one forest; add the rest to the other forest.
- Solve the two systems independently.

Finding the core

- Start with the GREEN edges in the core. If a vertex has two neighbours in the core add it to the core.
- If the graph is locally sparse, the process will stop.

Splitting the graph into forests

Nash-Williams Theorem: H: an undirected graph

$\forall X \subseteq V$ $(X \neq \emptyset)$ #edges induced by $X \leq 2(|X|-1)$.

Then, the edges of H can be partitioned into two forests.

Locally sparse graphs

G is locally (k,α)-locally sparse if every $V' \subseteq V$ with at most k vertices induces at most αk edges.

Existence of locally sparse graphs

Lemma: For all large N, there is a $(4N^{1/6}, 5/4)$- locally sparse graph with N vertices and $\Omega(N^{7/6})$ edges.

Pick each edge independently with prob. $p = \frac{1}{50} N^{-5/6}$.

$$\Pr[\text{Some } V' \subseteq V \text{ induces many edges}] \leq \sum_{\ell=4}^{4N^{1/6}} \binom{N}{\ell} \binom{\ell^2/2}{(5/4)\ell} p^{(5/4)\ell}$$

$$= o(1)$$

$$\mathbb{E}[\#\text{edges}] = p\binom{N}{2}$$

whp $\#\text{edge} \geq p\binom{N}{2}/2 = \Omega(N^{7/6})$.

Thus, $\exists$ $(4N^{1/6}, 5/4)$- sparse graph with $\Omega(N^{7/6})$ edges.

SUMMARY

- A lower bound

> For t a constant and $n \leq m$,
>
> $$S(m, n, t) \gtrsim t\, m^{1/t}\, n^{1-1/t}.$$

We used the polynomial method.

- A classical upper bound

$$S(m, n, t=2) = m^{1-1/n} \quad \text{for} \quad n = \pm 1 \pmod 3$$

$$= m^{1-1/(n-1)} \quad \text{for} \quad n = 0 \pmod 3$$

($S(m, n = 8, 9, t=2) = O(m^{6/7})$)

Using dense graphs of high girth.

- A quantum upper bound

$$Q\,S(m, n \leq m^{1/8}, t=2) = O(m^{7/8})$$

Using dense locally sparse graphs, Nash-Williams theorem

Questions

- What is $s(m,n,t)$?

  t a constant, $m \geq n \geq 1$

- What is $s^{\oplus}(m,n,t)$?

- Could even better schemes for $t=2$ be devised using XOR queries?

- We know $\Omega(m^{4/4}) \leq s(m,2,2) \leq O(m^{2/3})$.

  Can we get tight bounds?

Please see:

10.4230/LIPIcs.ICALP.2022.52

Thanks!