Quantum ordered search: Is $\frac{1}{\pi}\ln n$ the right answer?

Andrew Childs
University of Waterloo

Troy Lee
Columbia University

Ordered search problem

- Complexity of finding a given item in an ordered list.
- Given an ordered list $x_1 \leq x_2 \leq \ldots \leq x_n$ want to find position of given item z.
- Ask queries of the form $x_i \geq z$?
- How many queries are needed in worst case?

Formalization in standard query model

- Say that z is actually the i^{th} item in the list. Then answers to the query $x_j \geq z$ will look as follows: $0 \ldots 01 \ldots 1$.

- Thus can equivalently represent problem as querying bits of input and identifying first occurrence of a '1'.

- For example, for $n = 4$, set of inputs would be

$$S = \{1111, 0111, 0011, 0001\}.$$

  Note that last bit is always one.

- Problem is to identify the input (oracle identification problem).

Complexity of ordered search

- Classically, can succeed with $\log n$ queries by binary search and this is tight.
- In quantum case, one can do better. But only by a constant!
- Upper bounds: $0.631 \log n$ [HNS01], $0.526 \log n$ [FGGS99], $0.439 \log n$ [BJL04], $0.433 \log n$ [CLP06], $0.32 \log n$ [B-OH07] (bounded-error)
- Lower bounds: $\sqrt{\log n}/\log\log n$ [BW98], $\log n/\log\log n$ [FGGS98], $0.0833 \log n$ [Amb99], $\frac{1}{\pi}\ln n \approx 0.221 \log n$ [HNS01]
- What is this fundamental constant of quantum information?

Apologia

- Now it is clear we are talking about constant factors. But . . .
- Ordered search is a fundamental problem, and natural subroutine for sorting algorithms.
- On algorithm side, we still lack a good theoretical understanding.
- Lower bounds lead to some nice math.
- Would be really cool if the right answer is $\frac{1}{\pi}\ln n$.

This talk

- Describe how the problem can be simplified by symmetry arguments.
- Briefly discuss how current best exact algorithm is obtained.
- Main result: One of the best lower bound techniques, the adversary method, cannot show lower bounds larger than $\frac{1}{\pi}\ln n + O(1)$. Holds also for the "negative" adversary method [HLŠ07].

Symmetrization

- "Whenever you have to deal with a structure endowed entity Σ try to determine its group of automorphisms . . . you can expect to gain a deep insight into the constitution of Σ in this way."

  —Hermann Weyl, *Symmetry*

- For our purposes, an automorphism is a permutation τ that preserves agreement on the function:

$$f(x) = f(y) \iff f(\tau(x)) = f(\tau(y))$$

  for all x, y.

- But for original problem: $S = \{1111, 0111, 0011, 0001\}$ only have trivial automorphism.

Problem with cyclic structure

- [FGGS99] consider inputs of length $2n$ "on a circle":

  $S' = \{11110000, 01111000, 00111100, 00011110, 00001111, 10000111, 11000011, 11100001\}$

- Notice here that $x_i = 1 - x_{n+i}$. Second half is complement of first half.

- Complexity of this problem differs from that of the original by at most one query: If can solve problem with $2n$ inputs can also solve problem with n inputs as is subset. ∎

- Given algorithm for n input problem, first query x_n. If it is one, run algorithm on first half, otherwise run algorithm on second half.

Upper bounds

- Barnum, Saks, and Szegedy [BSS03] show that existence of a quantum t-query algorithm can be represented by a semidefinite program.

- Thus in principle we have an efficient way to compute quantum query complexity. In practice, however, it is often said that the BSS program is too complicated to be useful.

- In the case of ordered search, however, the symmetry of the problem allows the BSS program to be simplified greatly.

BSS program for ordered search

Find $2n$-by-$2n$ positive semidefinite matrices $M_i^{(j)}$ such that

$$\sum_{i=0}^{2n} M_i^{(0)} = E_0$$

$$\sum_{i=0}^{2n} M_i^{(j)} = \sum_{i=0}^{2n} E_i \circ M_i^{(j-1)}$$

$$\sum_{i=0}^{2n} M_i^{(t)} = I$$

where E_0 is the all ones matrix, and $E_i[x, y] = (-1)^{x_i + y_i}$.

Example: the matrix E_1

$$E_1 = \begin{array}{c@{}c} \begin{array}{cccccccc} 1111 & 1110 & 1100 & 1000 & 0000 & 0001 & 0011 & 0111 \end{array} & \\ \begin{bmatrix} 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ 1 & 1 & 1 & 1 & -1 & -1 & -1 & -1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \\ -1 & -1 & -1 & -1 & 1 & 1 & 1 & 1 \end{bmatrix} & \begin{array}{c} 1111 \\ 1110 \\ 1100 \\ 1000 \\ 0000 \\ 0001 \\ 0011 \\ 0111 \end{array} \end{array}$$

Binary search in the BSS framework

- Set $M_0^{(0)}, M_1^{(0)} = (1/2)E_0$ the all ones matrix. All other $M_i^{(0)}$ matrices will be zero.

- Then $M_0^{(0)} + M_1^{(0)} = E_0$, and

$$M_0^{(0)} + E_1 \circ M_1^{(0)} = \begin{array}{cc} \begin{array}{cccccccc} \rotatebox{90}{1111} & \rotatebox{90}{1110} & \rotatebox{90}{1100} & \rotatebox{90}{1000} & \rotatebox{90}{0000} & \rotatebox{90}{0001} & \rotatebox{90}{0011} & \rotatebox{90}{0111} \end{array} & \\ \begin{bmatrix} 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 1 & 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \\ 0 & 0 & 0 & 0 & 1 & 1 & 1 & 1 \end{bmatrix} & \begin{array}{l} 1111 \\ 1110 \\ 1100 \\ 1000 \\ 0000 \\ 0001 \\ 0011 \\ 0111 \end{array} \end{array}$$

Binary search in the BSS framework

We can continue, in this same way. Call the matrix from the last slide A. Setting $M_0^{(1)}, M_3^{(1)} = (1/2)A$, and all others zero, then $M_0^{(1)} + M_3^{(1)} = A$ as required and

$$M_0^{(1)} + E_3 \circ M_3^{(1)} = \begin{array}{cc} \begin{array}{cccccccc} \rotatebox{90}{1111} & \rotatebox{90}{1110} & \rotatebox{90}{1100} & \rotatebox{90}{1000} & \rotatebox{90}{0000} & \rotatebox{90}{0001} & \rotatebox{90}{0011} & \rotatebox{90}{0111} \end{array} & \\ \begin{bmatrix} 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 1 \end{bmatrix} & \begin{array}{c} 1111 \\ 1110 \\ 1100 \\ 1000 \\ 0000 \\ 0001 \\ 0011 \\ 0111 \end{array} \end{array}$$

Finally, with one more query we can reach the identity matrix.

Symmetrized program for ordered search

- The cyclical structure of the problem can be used to reduce the number of variable matrices to two for each query, one representing the null query $M_0^{(j)}$, and the other representing the query to the first bit $M_1^{(j)}$. The matrices $M_i^{(j)}$ for $i > 1$ will simply be permutations of $M_1^{(j)}$. ▮

- Childs, Landahl, and Parillo obtain the best exact algorithm by showing this program is feasible for $n = 605$ with 4 queries. Applying this algorithm recursively gives general upper bound of $4 \log_{605} n$.

Lower bounds: adversary method

- Main lower bound techniques: polynomial method and adversary method.
- Adversary method developed and improved in long series of works [BBBV94, Amb00, HNS01, BSS03, Amb03, LM04, Zha04, SŠ06, HLŠ07]
- Relation to BSS program: One can take the dual of the BSS program. By Farkas' lemma, the dual will be feasible iff the primal is infeasible. Thus one can show *lower bounds* by constructing solutions to the dual.
- The adversary bound implies solutions to the dual of a particular, restricted form.

Adversary method: matrix formulation

- Adversary bound is an optimization problem which can also be written as a semidefinite program.

$$\mathrm{ADV}(f) := \max_{\Gamma} \frac{\|\Gamma\|}{\max_i \|\Gamma \circ D_i\|}$$

where $\Gamma[x, y] = 0$ if $f(x) = f(y)$ and $D_i[x, y] = 1$ if $x_i \neq y_i$ and 0 otherwise.

- Symmetry also helps simplify the adversary bound. Automorphism principle [HLŠ07]: May assume without loss of generality, that optimal Γ satisfies $\Gamma[x, y] = \Gamma[\tau(x), \tau(y)]$ for every automorphism τ of f. Furthermore, if automorphism group is transitive, the uniform eigenvector will be a principal eigenvector of Γ and all $\|\Gamma \circ D_i\|$ are equal.

Γ matrix for OSP

$$\Gamma = \begin{array}{cc} \begin{array}{cccccccc} 1111 & 1110 & 1100 & 1000 & 0000 & 0001 & 0011 & 0111 \end{array} & \\ \begin{bmatrix} 0 & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & \gamma_3 & \gamma_2 & \gamma_1 \\ \gamma_1 & 0 & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & \gamma_3 & \gamma_2 \\ \gamma_2 & \gamma_1 & 0 & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & \gamma_3 \\ \gamma_3 & \gamma_2 & \gamma_1 & 0 & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \\ \gamma_4 & \gamma_3 & \gamma_2 & \gamma_1 & 0 & \gamma_1 & \gamma_2 & \gamma_3 \\ \gamma_3 & \gamma_4 & \gamma_3 & \gamma_2 & \gamma_1 & 0 & \gamma_1 & \gamma_2 \\ \gamma_2 & \gamma_3 & \gamma_4 & \gamma_3 & \gamma_2 & \gamma_1 & 0 & \gamma_1 \\ \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & \gamma_3 & \gamma_2 & \gamma_1 & 0 \end{bmatrix} & \begin{array}{c} 1111 \\ 1110 \\ 1100 \\ 1000 \\ 0000 \\ 0001 \\ 0011 \\ 0111 \end{array} \end{array}$$

Automorphism principle gives

$$\|\Gamma\| = \gamma_n + 2\sum_{i=1}^{n-1} \gamma_i.$$

$\Gamma \circ D_1$ matrix for OSP

$$\Gamma \circ D_1 = \begin{array}{c} \begin{array}{cccccccc} 1111 & 1110 & 1100 & 1000 & 0000 & 0001 & 0011 & 0111 \end{array} \\ \begin{bmatrix} 0 & 0 & 0 & 0 & \gamma_4 & \gamma_3 & \gamma_2 & \gamma_1 \\ 0 & 0 & 0 & 0 & \gamma_3 & \gamma_4 & \gamma_3 & \gamma_2 \\ 0 & 0 & 0 & 0 & \gamma_2 & \gamma_3 & \gamma_4 & \gamma_3 \\ 0 & 0 & 0 & 0 & \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 \\ \gamma_4 & \gamma_3 & \gamma_2 & \gamma_1 & 0 & 0 & 0 & 0 \\ \gamma_3 & \gamma_4 & \gamma_3 & \gamma_2 & 0 & 0 & 0 & 0 \\ \gamma_2 & \gamma_3 & \gamma_4 & \gamma_3 & 0 & 0 & 0 & 0 \\ \gamma_1 & \gamma_2 & \gamma_3 & \gamma_4 & 0 & 0 & 0 & 0 \end{bmatrix} \end{array} \begin{array}{c} 1111 \\ 1110 \\ 1100 \\ 1000 \\ 0000 \\ 0001 \\ 0011 \\ 0111 \end{array}$$

- We see that $\|\Gamma \circ D_1\| = \|\mathrm{Toeplitz}(\gamma_n, \ldots, \gamma_1)\|$.

Høyer, Neerbeck, Shi construction

Assume that n is even. Let $\gamma_i = 1/i$ for $i = 1, \ldots, n/2$ and zero otherwise. Then objective function is

$$2\sum_{i=1}^{n/2} \frac{1}{i} \approx 2\ln(n/2)$$

and have to upper bound spectral norm of

$$\begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 & 1/2 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1/2 \\ 0 & 0 & 0 & 0 & 1/2 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 1/2 & 0 & 0 \\ 0 & 0 & 1/2 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1/2 & 0 & 0 & 0 & 0 \\ 1/2 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 1 & 1/2 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}.$$

“Half” Hilbert matrix

In general, spectral norm of $\Gamma_{2n} \circ D_1$ will be given by spectral norm of

$$Z_n = \begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 & \dots & 1/n \\ 1/2 & 1/3 & 1/4 & \dots & 1/n & 0 \\ 1/3 & 1/4 & \dots & 1/n & 0 & 0 \\ \vdots & \dots & & & \vdots & \vdots \\ 1/(n-1) & 1/n & 0 & 0 & 0 & 0 \\ 1/n & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

Hilbert's Inequality

Consider the "full" Hilbert matrix

$$H = \begin{pmatrix} 1 & 1/2 & 1/3 & 1/4 & \dots \\ 1/2 & 1/3 & 1/4 & \dots & \dots \\ 1/3 & 1/4 & \dots & & \dots \\ 1/4 & \dots & & & \vdots \\ \vdots & & & \vdots & \ddots \end{pmatrix}$$

Hilbert showed (with improvement by Schur) that $\|H\| \leq \pi$. Thus HNS construction gives

$$\mathrm{ADV}(\mathrm{OSP}_n)) \geq \frac{2\ln(n/2)}{\pi}.$$

General question

This construction raises the following question: Given a matrix of the form

$$A_n = \begin{pmatrix} a_0 & a_1 & a_2 & a_3 & \dots & a_{n-1} \\ a_1 & a_2 & a_3 & \dots & a_{n-1} & 0 \\ a_2 & a_3 & \dots & a_{n-1} & 0 & 0 \\ \vdots & \dots & & & \vdots & \vdots \\ a_{n-2} & a_{n-1} & 0 & 0 & 0 & 0 \\ a_{n-1} & 0 & 0 & 0 & 0 & 0 \end{pmatrix}$$

how large can $\sum_i a_i$ be while $\|A_n\| \leq 1$? Let $\alpha(n)$ represent this optimal value.

Answer

For the case of non-negative matrices, we are able to give the exact answer:

$$\alpha^+(n) = \sum_{i=0}^{n-1} \left(\frac{\binom{2i}{i}}{4^i} \right)^2 = \frac{1}{\pi}(\ln n + \gamma + \ln 8) + O(1/n)$$

and explicit matrices which realize this bound.

Note that

$$\frac{\binom{2i}{i}}{4^i} \approx \frac{4^i/\sqrt{\pi i}}{4^i} = \frac{1}{\sqrt{\pi i}}.$$

Application to adversary bound

Turns out that this construction is also optimal for the adversary bound. The dual of the (non-negative) adversary bound is the following:

$$\min \mathrm{Tr}(P) \text{ subject to } P \succeq 0, \mathrm{tr}_i(P) \geq 1 \text{ for } i = 0, \ldots, n-1.$$

We exhibit a solution of this with the same value to show that

$$\mathrm{ADV}^+(\mathrm{OSP}_{2n}) = 2\alpha^+(n)$$

In the case of negative entries—with much more work—can show

$$\mathrm{ADV}(\mathrm{OSP}_n) \leq \mathrm{ADV}^+(\mathrm{OSP}_{2n}) + 1.$$

A word about the proof (non-negative case)

- We exhibit solutions to both the primal and dual formulation of adversary bound, and show that they match.

- A key role in both directions is played by the lovely sequence

$$\beta_i = \frac{\binom{2i}{i}}{4^i}.$$

- Key property: $\sum_{i=0}^{j} \beta_i \beta_{j-i} = 1$ for every j.

- Proof:

$$\frac{1}{\sqrt{1-z}} = \beta_0 + \beta_1 z + \beta_2 z^2 + \beta_3 z^3 + \dots$$

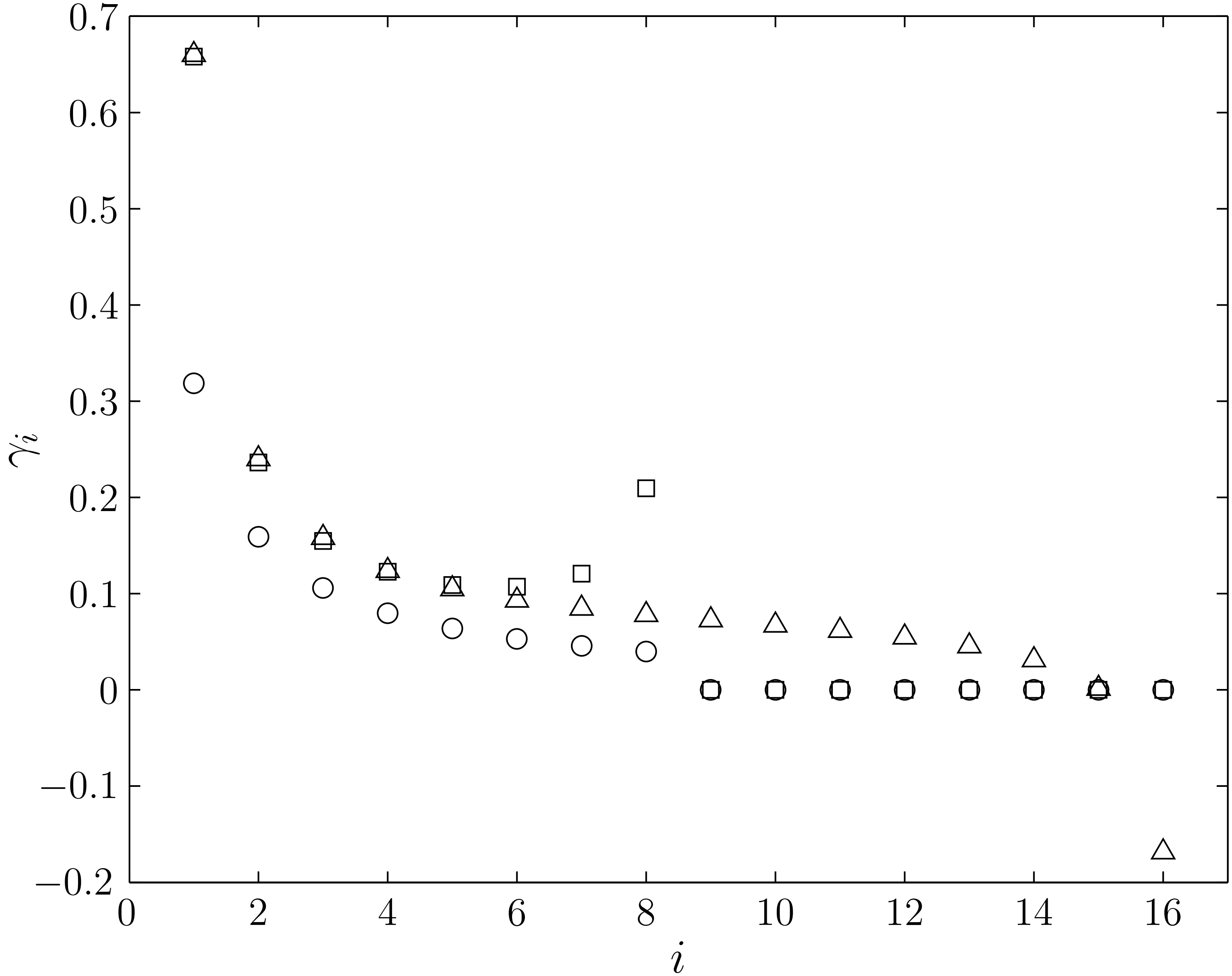
Optimal matrix (lower bound)

Recall we wish to show that $\alpha^{+}(n) \geq \sum_{i=0}^{n-1} \left(\frac{\binom{2i}{i}}{4^i} \right)^2$.

Define $A_n(j) = \sum_{i=0}^{n-j-1} \beta_i \beta_{i+j}$.

$$\begin{pmatrix} A_4(0) - A_4(1) & A_4(1) - A_4(2) & A_4(2) - A_4(3) & A_4(3) \\ A_4(1) - A_4(2) & A_4(2) - A_4(3) & A_4(3) & 0 \\ A_4(2) - A_4(3) & A_4(3) & 0 & 0 \\ A_4(3) & 0 & 0 & 0 \end{pmatrix}$$

To bound spectral norm, show that $x = [\beta_3, \beta_2, \beta_1, \beta_0]$ is eigenvector with eigenvalue 1. As x is non-negative and matrix is symmetric and non-negative, this must correspond to largest eigenvalue.


0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
−0.1
−0.2
γ_i
0
2
4
6
8
10
12
14
16
i

Conclusion

- Progress on ordered search will require new algorithms or new lower bound techniques.

- We have a solution to the dual BSS program which (I believe) is asymptotically optimal. Can one use sufficiency conditions for optimality of solutions to semidefinite programs to show this is the case?

- Observed with Peter Høyer: Our optimal matrix can be used to give nearly elementary proof of Hilbert's Inequality (need $\Gamma(1/2) = \sqrt{\pi}$).