UNIVERSITY
OF COLOGNE

LOWER BOUNDS USING QUANTUM QUERY COMPLEXITY

Speaker: Tristan Kahl

Tutor: Shahrukh Chishti

Institut für Theoretische Physik

2nd July 2026

0

OVERVIEW

Overview

- Motivation
- What are O, Θ and Ω?
- Classical lower bounds
- Quantum lower bounds
- Example: Grover's search

Motivation

- Compare classical and quantum computing

  → How much better ist a quantum computer?
- Find optimal quantum algorithms (and know when it is optimal)

  → Is there something faster than Grover's search?
- Understand limits of quantum computing

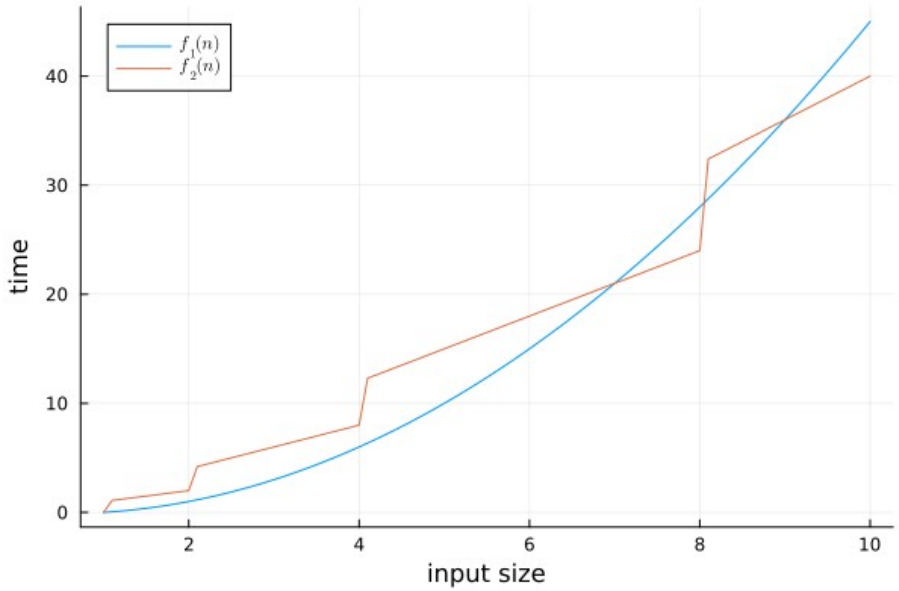
  → Will cryptography still work?

1

MATHEMATICAL CONCEPTS

📈 Big-O-Notation / Bachmann-Landau-Notation

- Basic idea: measure cost (time) as function of input size
- How can we do this?
  → Define some step: classical key comparison / oracle query / model like Turing machine or DFA / ...
  → Estimate number of steps
- Why do we ignore constants?
  → Example: $f_1(n) \sim n^2$, $f_2(n) \sim n\ log(n)$
  We see: for large n constants do not matter



📈 Big-O-Notation / Bachmann-Landau-Notation

- O: upper bound $\exists_{C>0}\exists_{m>0}\forall_{n>m}\ |f(n)| \leq C \cdot |g(n)|$
- Ω: lower bound $\exists_{c>0}\exists_{m>0}\forall_{n>m}\ c \cdot |g(n)| \leq |f(n)|$
- Θ: exact bound $\exists_{c,C>0}\exists_{m>0}\forall_{n>m}\ c \cdot |g(n)| \leq |f(n)| \leq C \cdot |g(n)|$
- Example 1: search in unsorted (classical) database of size N
  → worst case N steps ⇒ f ∈ O(N)
- Example 2: Bubble Sort
  → Count key comparisons
  → $\Sigma_n \Sigma_i 1 = \Sigma_n (n-1) = n(n-1)/2 \Rightarrow f \in O(N^2)$
- Constants do not matter:
  $O(N(N-1)/2) = O(N^2/2-N/2) = O(N^2/2) = O(N^2) \subset O(poly(N))$

```
bubbleSort(Array A)
  for (n = A.size; n > 1; n = n - 1) {
    for (i = 0; i < n - 1; i = i + 1) {
      if (A[i] > A[i + 1]) {
        A.swap(i, i + 1)
      }
    }
  }
```

2

LOWER BOUNDS: CLASSICAL AND BASIC IDEA

🏛 Classical lower bounds

- Idea: algorithm must decide between N different inputs
- Example:
  - Sorting: N = n! possible inputs
  - Every step two possible ways ⇒ $2^{f(n)}$ different cases covered
  - $2^{f(n)} \geq n!$
    ⇒ $f(n) \geq \log(n!) \approx n/e \log(n/e)$
    ⇒ $f(n) \in \Omega(n \log n)$

    ⇒ optimal sorting algorithm takes n log n steps

Quantum lower bounds

- Quantum query algorithm: model focusing on queries / number of oracle calls
- Idea: measure entanglement
- Main questions:
  - How many queries are needed for some entanglement?
  - How much entanglement is needed for a working algorithm?
- $U_0 \rightarrow O \rightarrow U_1 \rightarrow O \rightarrow \ldots \rightarrow U_T$

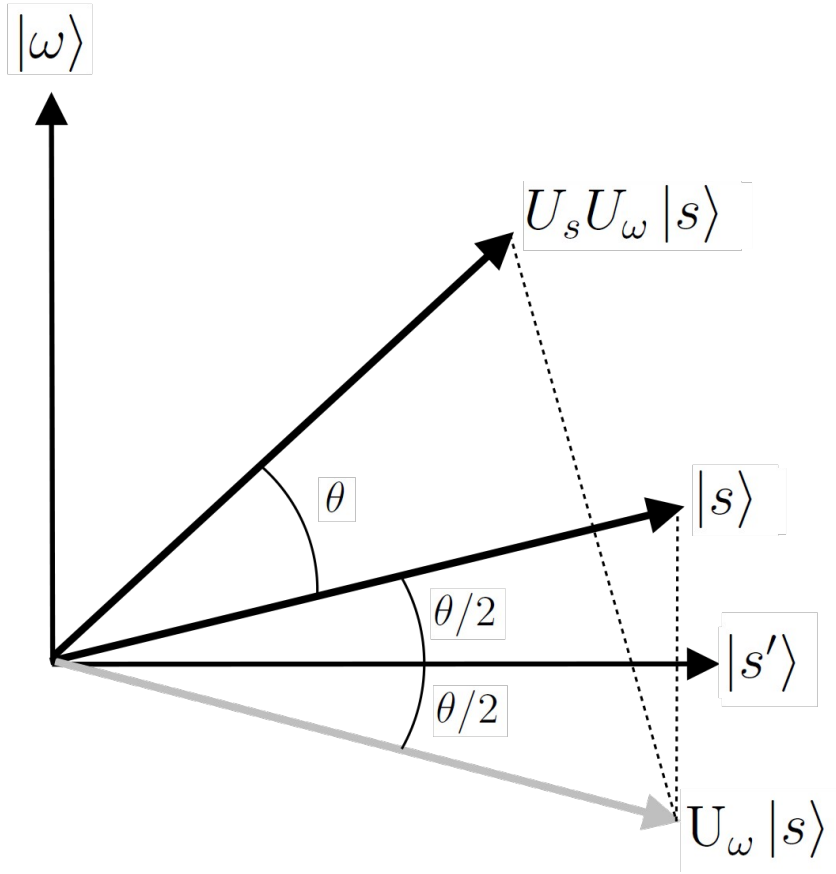
  O: oracle transformations
  U_j: arbitrary unitary operations not depending on input bits
- What is an Oracle? → black box, $O|x\rangle = (-1)^{f(x)}|x\rangle$ for a binary function f and basis states $|x\rangle$

Recap: Grover's Algorithm (Upper Bound)

- Search in unorderes database with N entries
- Function f: {0, 1, …, N−1} → {0, 1} returning 1 if x satisfies criterion

  → can be accessed via oracle: $U_w |x\rangle = (-1)^{f(x)}|x\rangle$
- Algorithm

  1. Initialization: $|s\rangle = 1/\sqrt{N} \Sigma |x\rangle$
  2. Grover iteration:
     → Oracle U_w
     → Grover diffusion $U_s = 2 |s\rangle\langle s| - I$
  3. Measuring state



Wikipedia contributors. (2026, June 22). Grover's algorithm.

3

QUANTUM LOWER BOUNDS

Quantum lower bounds – main idea

- Bipartite system: $H = H_A \otimes H_I$
  - H_A: workspace
  - H_I: input space
- $|\psi_{start}\rangle = |0\rangle \otimes \Sigma\, \alpha_x\, |x\rangle$, not entangled
  choose for example $\alpha_x = 1/\sqrt{m}$ (equal superposition)
- $|\psi_{end}\rangle = \Sigma\, \alpha_x\, |\psi_x\rangle \otimes |x\rangle$, must be entangled (Why? For every algorithm?)
- Example: $1/\sqrt{m}\ \Sigma\, |x\rangle|\varphi_x\rangle \otimes |x\rangle$ reproduces input and is entangled
- How do we measure entanglement?
  → Entanglement entropy $S = -\,\mathrm{tr}\,\rho \log \rho$
  → Or coherences (sum of all non diagonal entries)
- $\rho_{end} = \mathrm{tr}_A\, |\psi_{end}\rangle\langle\psi_{end}|$

Quantum lower bounds – Lemma 1

- **Claim:** Let A be an algorithm that computes f with probability at least $1 - \varepsilon$. Let x, y be such that $f(x) \neq f(y)$. Then, $|(\rho_{end})_{xy}| \leq 2\sqrt{\varepsilon(1-\varepsilon)}\,|\alpha_x||\alpha_y|$.
  → compare to $|(\rho_{start})_{xy}| = |\alpha_x||\alpha_y| \Rightarrow$ non-diagonal entries get lower
  → special case $\varepsilon = 0 \Rightarrow \rho_{end}$ is diagonal
- **Proof:** Express $|\Psi_x\rangle$ as $\Sigma\, a_{z,v}\,|z\rangle|v\rangle$ and $|\Psi_y\rangle$ as $\Sigma\, b_{z,v}\,|z\rangle|v\rangle$
  where z is the answer part.

  We get $|(\rho_{end})_{xy}| = |\alpha_x||\alpha_y| \cdot |\Sigma\, a_{z,v}\, b_{z,v}| \leq |\alpha_x||\alpha_y| \cdot \Sigma\, |a_{z,v}|\,|b_{z,v}| = |\alpha_x||\alpha_y| \cdot (\Sigma_{z=f(x)}\,|a_{z,v}|\,|b_{z,v}| + \Sigma_{z\neq f(x)}\,|a_{z,v}|\,|b_{z,v}|)$

  $= |\alpha_x||\alpha_y| \cdot (\sqrt{(\Sigma_{z=f(x)}\,|a_{z,v}|^2)}\,\sqrt{(\Sigma_{z=f(x)}\,|b_{z,v}|^2)} + \sqrt{(\Sigma_{z\neq f(x)}\,|a_{z,v}|^2)}\,\sqrt{(\Sigma_{z\neq f(x)}\,|b_{z,v}|^2)})$
- $= \sqrt{((1-\varepsilon_1)\varepsilon_2)}\,|\alpha_x||\alpha_y| + \sqrt{(\varepsilon_1(1-\varepsilon_2))}\,|\alpha_x||\alpha_y| \leq 2\sqrt{\varepsilon(1-\varepsilon)}\,|\alpha_x||\alpha_y|$.
- What did we get?
  → Constraints on density matrix: Algorithm must destroy coherence
  next step: bound change of density matrix per query

Example: Unordered search – Theorem 1 (→ Grover)

- **Claim:** Any quantum algorithm that finds i with probability $1 - \varepsilon$ uses $\Omega(\sqrt{N})$ queries.
- **Idea of Proof:** ρ_k density matrix after k queries, S_k sum of all absolute values of non-diagonal entries

  1. $S_0 = N-1$ (→ $N(N-1)$ non-diagonal entries)
  2. $S_T \leq 2\sqrt{\varepsilon(1-\varepsilon)}\,(N-1)$ (→ Lemma 1)
  3. $S_{k-1} - S_k \leq 2\sqrt{N-1}$

- $S_0 - S_1 + S_1 - S_2 + \ldots + S_{T-1} - S_T = S_0 - S_T \geq (1 - 2\sqrt{\varepsilon(1-\varepsilon)})\,(N-1)$

  $\Rightarrow (1 - 2\sqrt{\varepsilon(1-\varepsilon)})\,(N-1) \leq 2\,T\sqrt{N-1} \Rightarrow T \geq (1 - 2\sqrt{\varepsilon(1-\varepsilon)})\sqrt{N-1}/2$

  $\Rightarrow \Omega(\sqrt{N})$

4

SUMMARY

Summary

- Correctly working algorithms must create entanglement

  ⇒ Lower bound on computation resources
- Grover's algorithm is optimal

  ⇒ No faster quantum algorithm possible
- Search problem can be extended → universal bounds for NP-problems

References

- Ambainis: "Quantum Lower Bounds by Quantum Arguments" https://arxiv.org/abs/quant-ph/0002066
- Wikipedia contributors. (2026, June 17). NP-hardness. In Wikipedia, The Free Encyclopedia. Retrieved 13:23, July 1, 2026, from https://en.wikipedia.org/w/index.php?title=NP-hardness&oldid=1359837082
- Wikipedia contributors. (2026, June 22). Grover's algorithm. In Wikipedia, The Free Encyclopedia. Retrieved 13:25, July 1, 2026, from https://en.wikipedia.org/w/index.php?title=Grover%27s_algorithm&oldid=1360680967