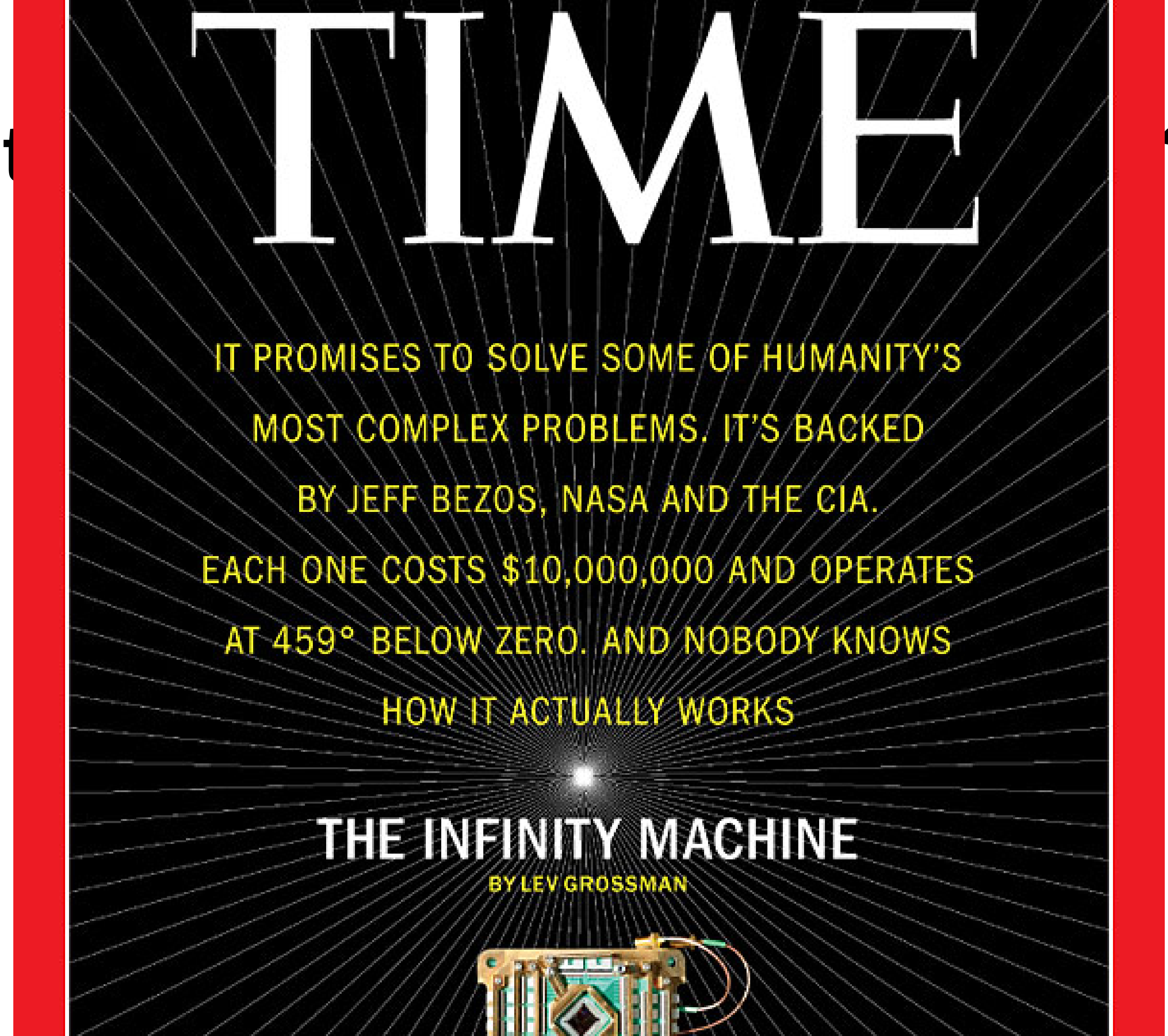
Problems with Multiple Oracles

Shelby Kimmel

Edward Farhi

Center for Theoretical Physics, MIT

What t … ?



Time magazine
Feb 17, 2014

What to do with a Quantum Computer?

- Let’s try to solve something hard: 3-SAT

Traveling Salesman Problem
[Karp ‘72]

Tetris
[Demaine et al ‘03]

Scheduling jobs
[Ullman ‘75]

Is a graph planar?
[Grigoriev et al ‘07]

Graph Coloring
[Karp ‘72]

Sudoku
[Yato et al]

3-SAT

- Goal: Want to satisfy a set of Boolean clauses, each with 3 variables.

$$(x_1 \lor \neg x_2 \lor x_3) \land (x_2 \lor x_3 \lor \neg x_4)$$

Each variable x_i can take value 0 or 1

$\lor$ is logical OR:

$0 \land 0 = 0$

$0 \land 1 = 1$

$1 \land 0 = 1$

$1 \land 1 = 1$

$\land$ is logical AND:

$0 \land 0 = 0$

$0 \land 1 = 0$

$1 \land 0 = 0$

$1 \land 1 = 1$

$\neg$ is logical NOT:

$\neg 0 = 1$

$\neg 1 = 0$

3-SAT

Is there an assignment of the n variables $\{x_1, x_2, \cdots, x_n\}$ such that $F(x_1, x_2, \cdots, x_n) = 1$, where

$$F(x_1, x_2, \cdots, x_n) = (x_1 \vee \neg x_3 \vee x_7) \wedge (x_2 \vee x_3 \vee \neg x_6) \wedge (x_1 \vee x_7 \vee x_{10}) \ldots$$

~poly(n) clauses (e.g. Cn^2)

Algorithm for 3-SAT

$$F(x_1, x_2, \cdots, x_n) = (x_1 \vee \neg x_3 \vee x_7) \wedge (x_2 \vee x_3 \vee \neg x_6) \wedge (x_1 \vee x_7 \vee x_{10}) \ldots$$

$\sim$poly(n) clauses (e.g. Cn^2)

- Guess a satisfying assignment. Test if all clauses are satisfied
  - Need to test $\sim 2^n$ possible inputs. With quantum computer can do in $\sqrt{2^n}$ steps

Outline

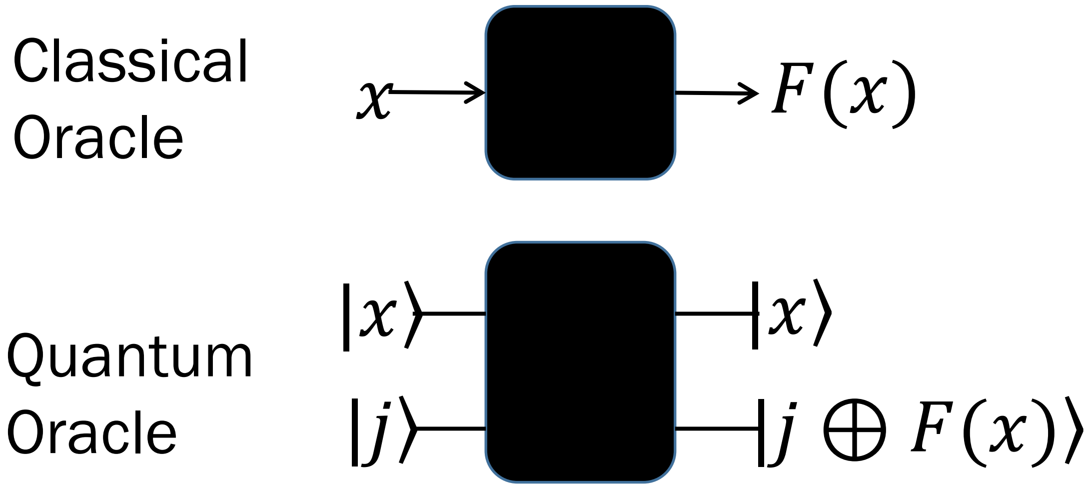
- Oracles and Oracle Models
- Related work
- Simple Example: Search with Multiple Oracles
- Open Problems and Directions for Future Work

Standard Oracle Model vs 3-SAT

3-SAT	Oracle Model
Given a function $F(x_1, x_2, \cdots, x_n) = (x_1 \vee \neg x_3 \vee x_7)$	Initially function F is unknown
Determine if input x exists such that $F(x) = 1$	Determine some property of F • Does x exists such that $F(x) = 1$? • Is F one-to-one?

Standard Oracle Model

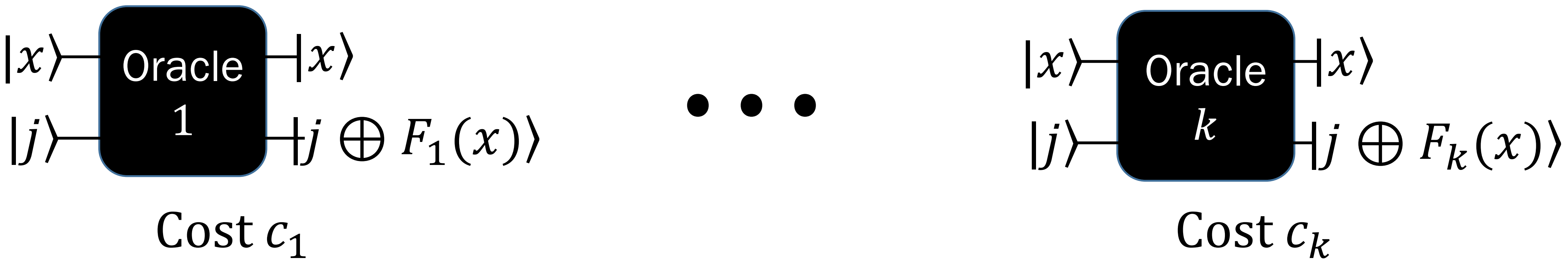
Goal: Determine a property of a function $F(x)$ for Boolean input $x = \{x_1, x_2, \cdots, x_n\}$, given an oracle for F.



Only care about # of oracle calls (queries)

Multiple Oracles with Costs Model

Goal: Determine a property of a function $F(x)$ for Boolean input $x = \{x_1, x_2, \cdots, x_n\}$, given a set of oracles associated with functions $\{F_1, \cdots, F_k\}$ which each have some information related to F



Care about total cost $= \sum_{i=1}^{k} q_i c_i$ where q_i is the # of times Oracle i is used

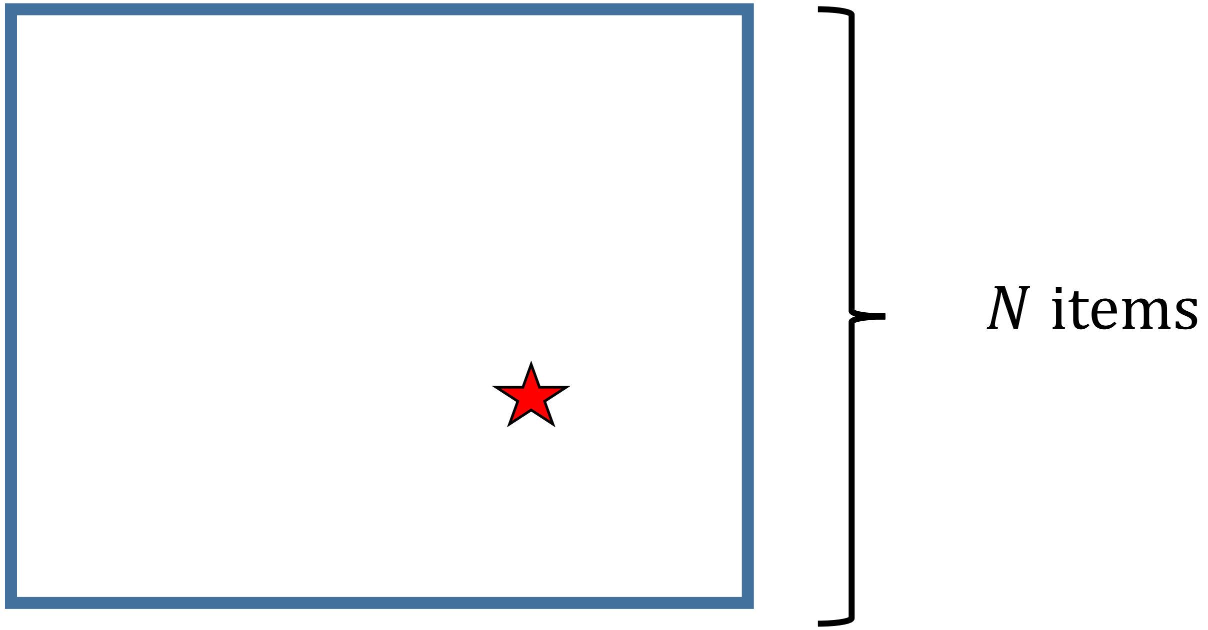
Related Work

- Ambainis '10: One oracle, but querying different x requires different amounts of time
  - E.g. To learn $f(00\cdots00)$ takes time 1, but to learn $f(11\cdots11)$ takes time 2
- Montanaro '09: Searching when given some additional information as to the location of the marked item.
  - E.g. Told that $f(00\cdots00)=1$ is more likely than $f(11\cdots11)$
- Cerf et al. '00: Use multiple oracles to speed up evaluation of satisfiability problems.
  - No cost, No lower bounds, Need certain structure.

Searching with an Oracle

Goal: Determine x such that $F(x) = 1$. Can ask oracle, "Is the i^{th} item the starred item?"

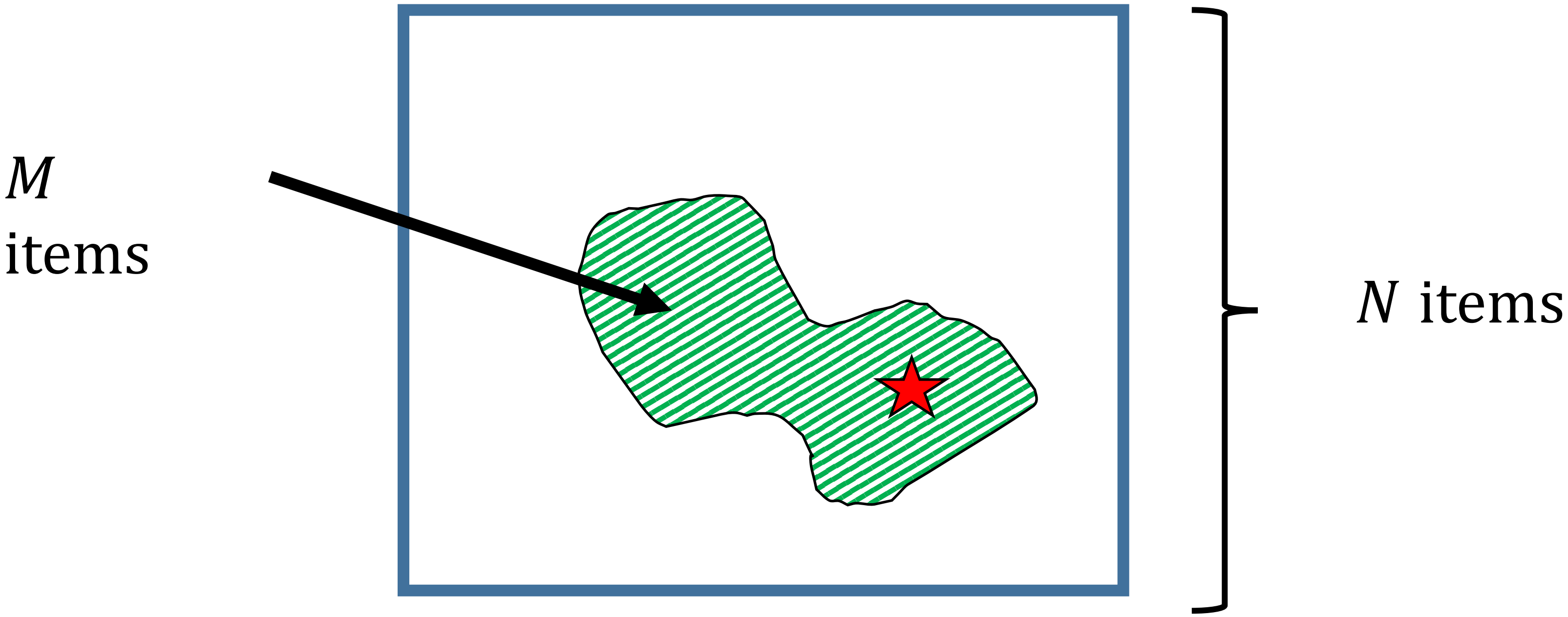
- Classically: $\sim N$ queries to oracle
- Quantumly: $\sim\sqrt{N}$ queries to oracle [Grover '97, Bennett et al. '97, Zalka '99]



Searching with Multiple Oracles

Can ask ★ **Oracle**, "Is the i^{th} item starred?"
Can ask **Oracle**, "Is the i^{th} item striped?"

Promised: The starred item is also striped

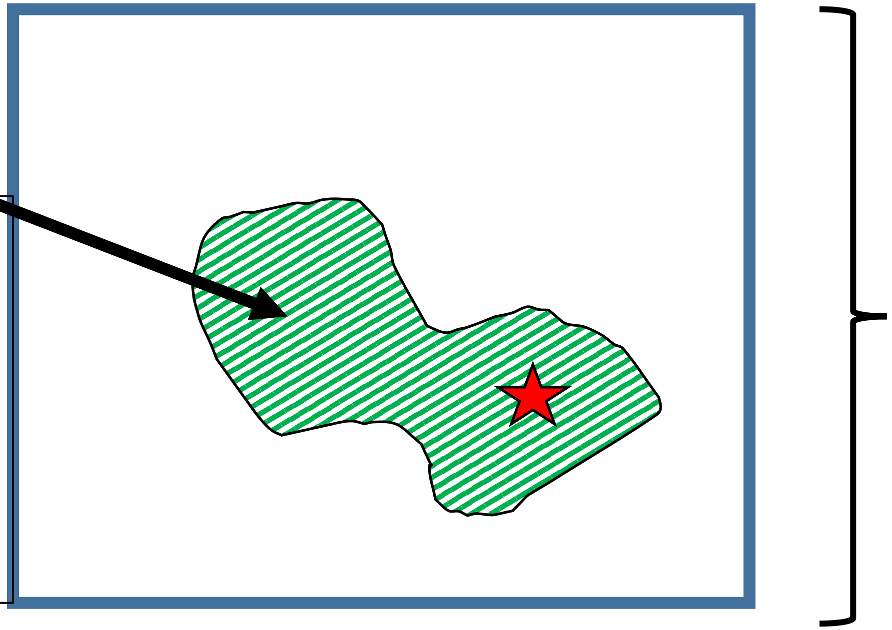


Searching with Multiple Oracles

Can ask **Oracle**, “Is the i^{th} item starred?”
Can ask **Oracle**, “Is the i^{th} item striped?”

Promised: The starred item is also striped

M items



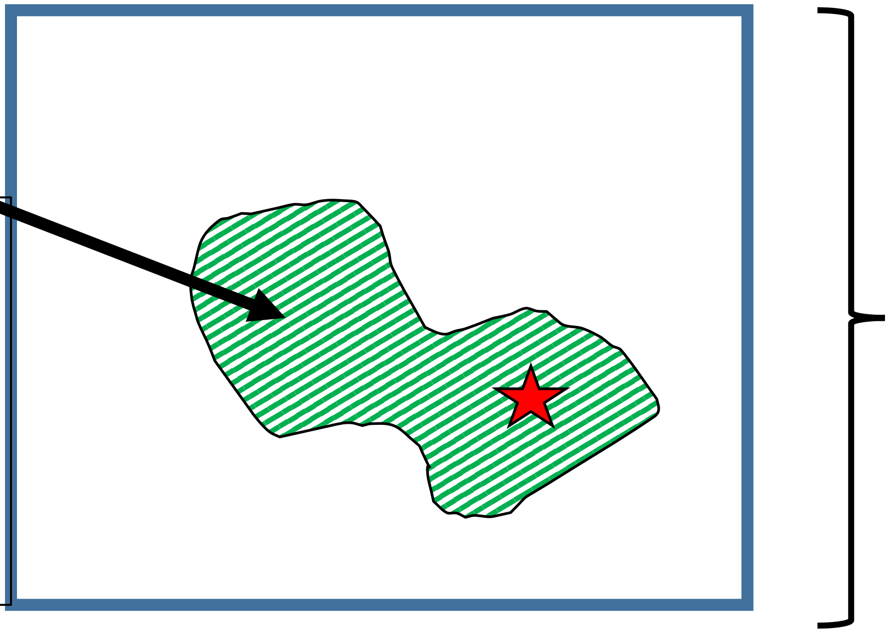
N items

Can you find the starred item with fewer total oracle queries using **Oracle**?
…

Searching with Multiple Oracles

Can ask **Oracle**, “Is the i^{th} item starred?”
Can ask **Oracle**, “Is the i^{th} item striped?”

Promised: The starred item is also striped



Can you find the starred item with fewer total oracle queries using **Oracle**?

… **NO**

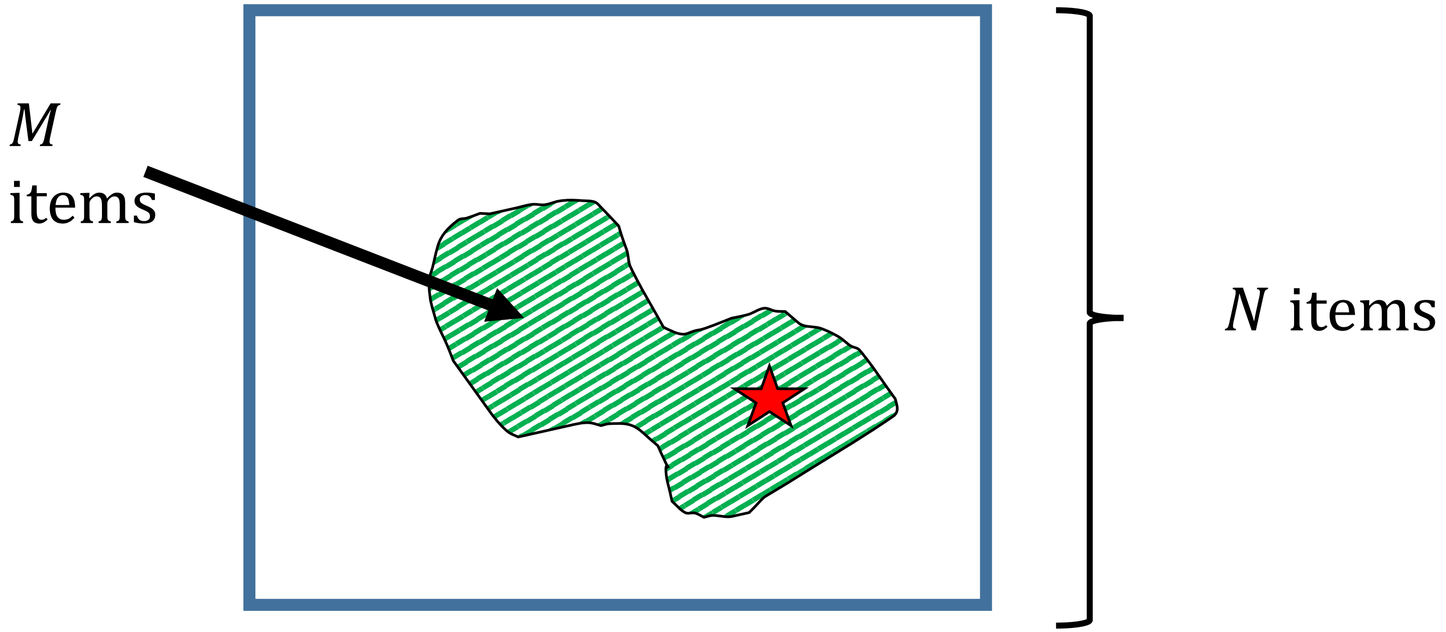
Searching with Multiple Oracles

Can ask ★ **Oracle**, "Is the i^{th} item starred?" **with cost $c_\star$**

Can ask ⦀ **Oracle**, "Is the i^{th} item striped?" **with cost $c_\parallel$**

$c_\star > c_\parallel$

Promised: The starred item is also striped



Searching with Multiple Oracles

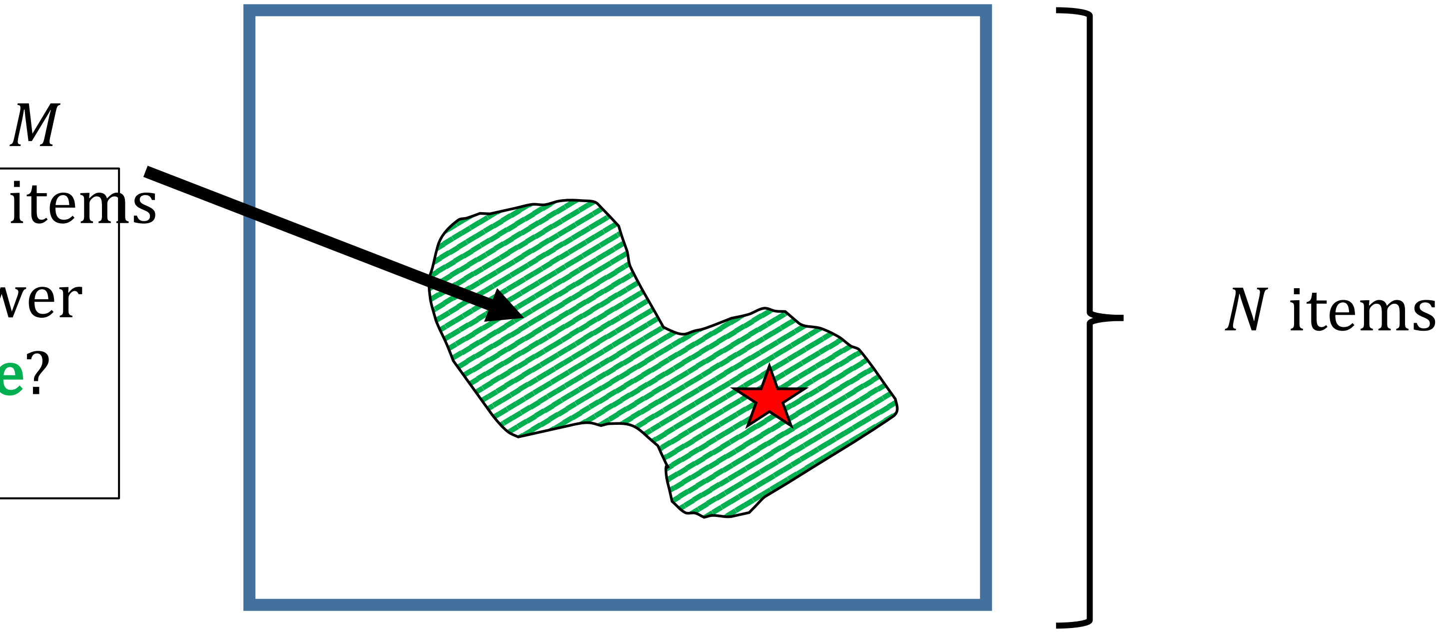
Can ask ★ **Oracle**, "Is the i^{th} item starred?" **with cost** $c_\star$

Can ask ◍ **Oracle**, "Is the i^{th} item striped?" **with cost** $c_\|$

$c_\star > c_\|$

Promised: The starred item is also striped

Can you find the starred item at lower cost using ◍ **Oracle**?
...



Searching with Multiple Oracles

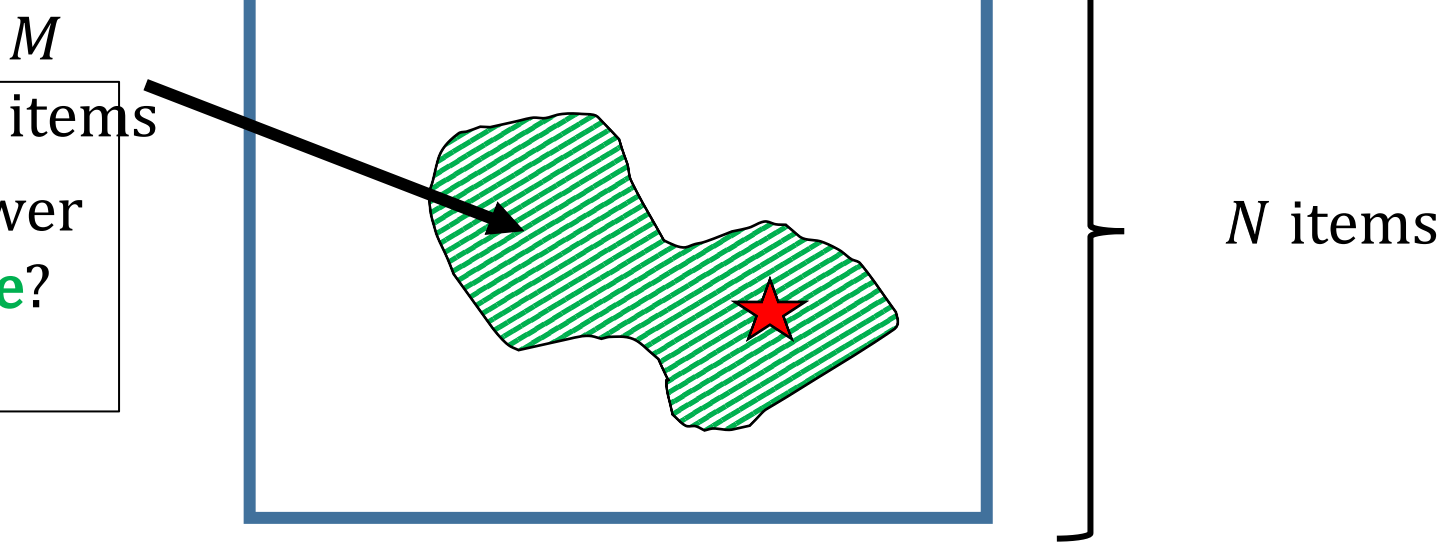
Can ask ★ Oracle, “Is the i^{th} item starred?” **with cost** $c_\star$

Can ask Oracle, “Is the i^{th} item striped?” **with cost** $c_\parallel$

$c_\star > c_\parallel$

Promised: The starred item is also striped

Can you find the starred item at lower cost using Oracle?
… **YES**



Searching with Multiple Oracles

Can ask ★ Oracle, "Is the i^{th} item starred?" **with cost $c_\star$**

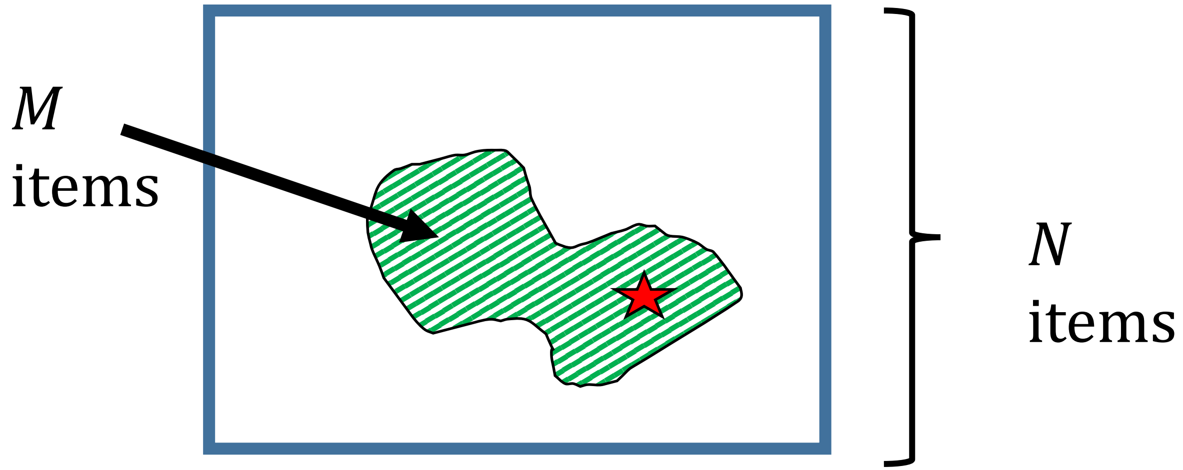
Can ask ◍ Oracle, "Is the i^{th} item striped?" **with cost $c_\parallel$**

$c_\star > c_\parallel$

Promised: The starred item is also striped

Classical ~ $\min\{c_\star N,\quad c_\parallel N + c_\star M\}$

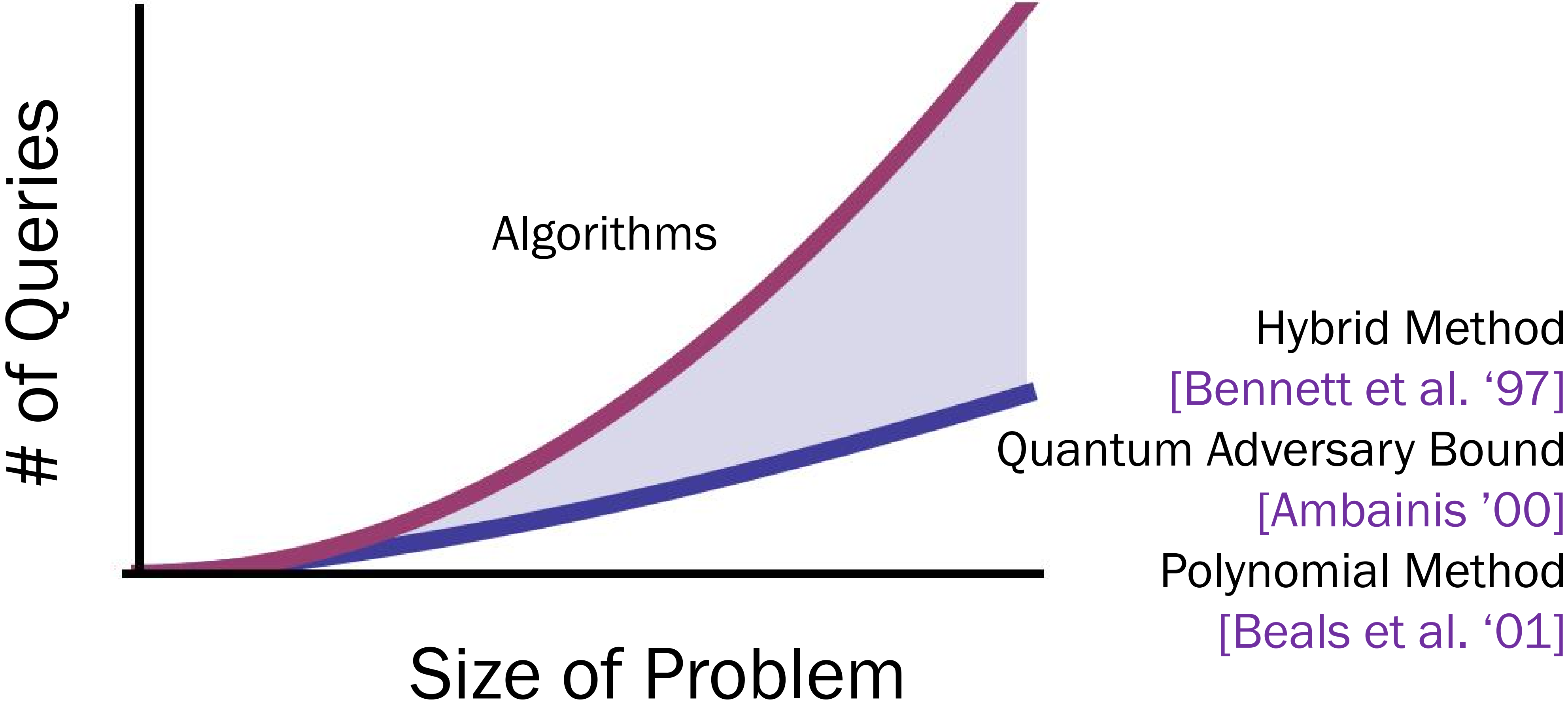
Quantum ~ $\min\{c_\star\sqrt{N},\ c_\parallel\sqrt{N} + c_\star\sqrt{M}\}$



Sometimes best to check all N items using ★Oracle

Otherwise can check all N items using cheaper oracle, but still need to use ★Oracle for M items

Quantum Query Complexity Bounds



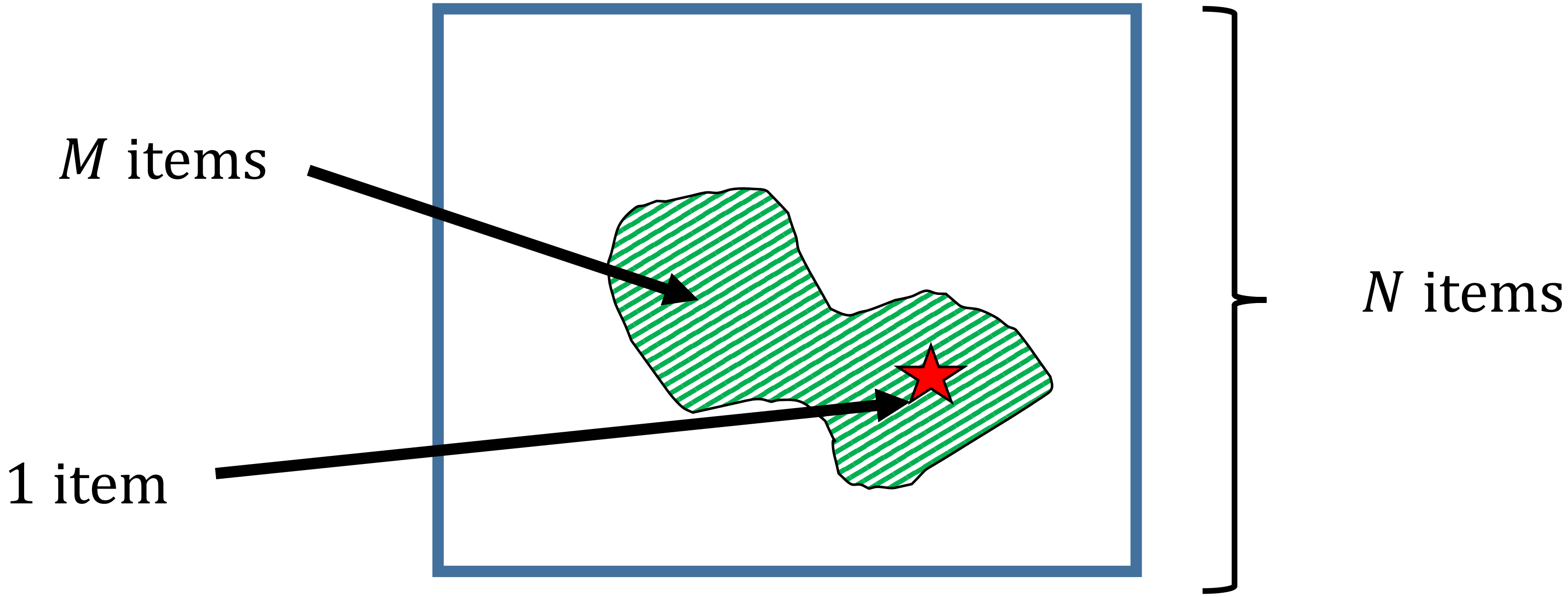
Lower Bounds for Search with Multiple Oracles

- **TOOL:**
  - Need at least $\sim\sqrt{N}$ queries to quantum oracle to find one item out of N [Bennett et al '97]

* "at least $\sim\sqrt{N}$" means at least $A\sqrt{N}$ for some constant A as N gets large. $= \Omega(\sqrt{N})$

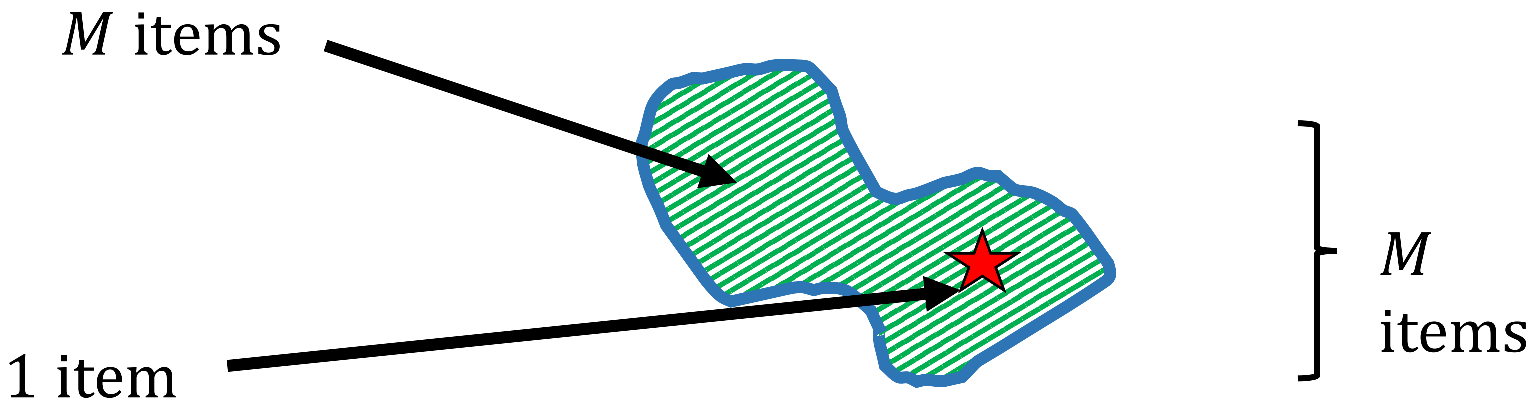
Lower Bounds for Search with Multiple Oracles

- How much can **Oracle** help?



Lower Bounds for Search with Multiple Oracles

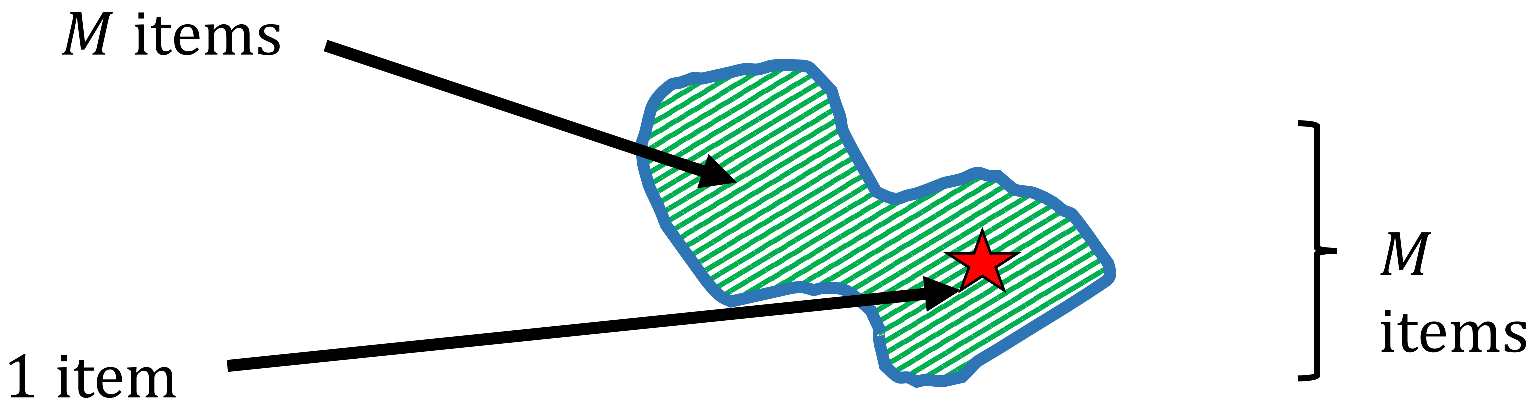
- How much can Oracle help?



Lower Bounds for Search with Multiple Oracles

- How much can Oracle help?

➢ At best, can narrow down to M items. But now searching for 1 among M, need at least $\sim\sqrt{M}$ queries to ★ **Oracle** TOOL



Lower Bounds for Search with Multiple Oracles

- How much can Oracle help?

➢ At best, can narrow down to M items. But now searching for 1 among M, need at least $\sim\sqrt{M}$ queries to ★ **Oracle** **TOOL**

1.

★ Oracle has cost $c_\star$, and need to use it $\sim\sqrt{M}$ times.
Always will have a cost of $\sim c_\star\sqrt{M}$

Lower Bounds for Search with Multiple Oracles

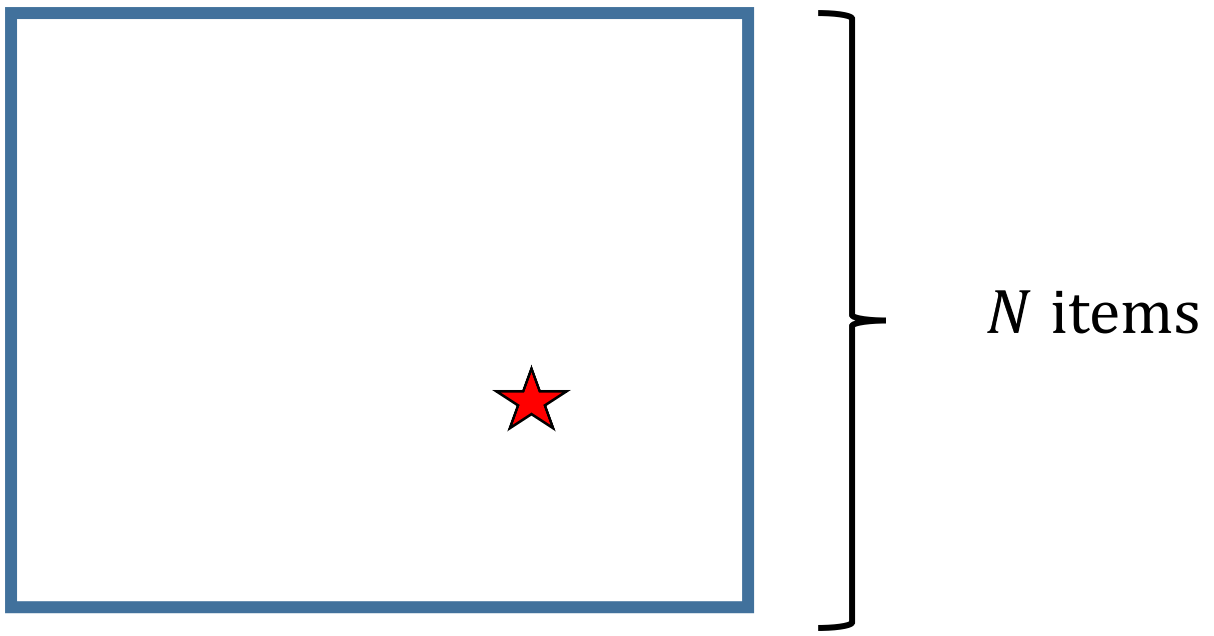
- Need ★ **Oracle** $\sim\sqrt{M}$ times; How many times do we need **Oracle**?
- What if only needed to use **Oracle** $\sqrt{M}$ times? (For contradiction.)
  - This would be awesome!
  - Would always want to have an **Oracle**, because it helps so much

Idea: Even if aren't given **Oracle,** create it using ★ **Oracle**.

Lower Bounds for Search with Multiple Oracles

- What if only needed to use Oracle $\sqrt{M}$ times? (For contradiction.)

Idea: Even if aren’t given Oracle, create it using ★ Oracle.

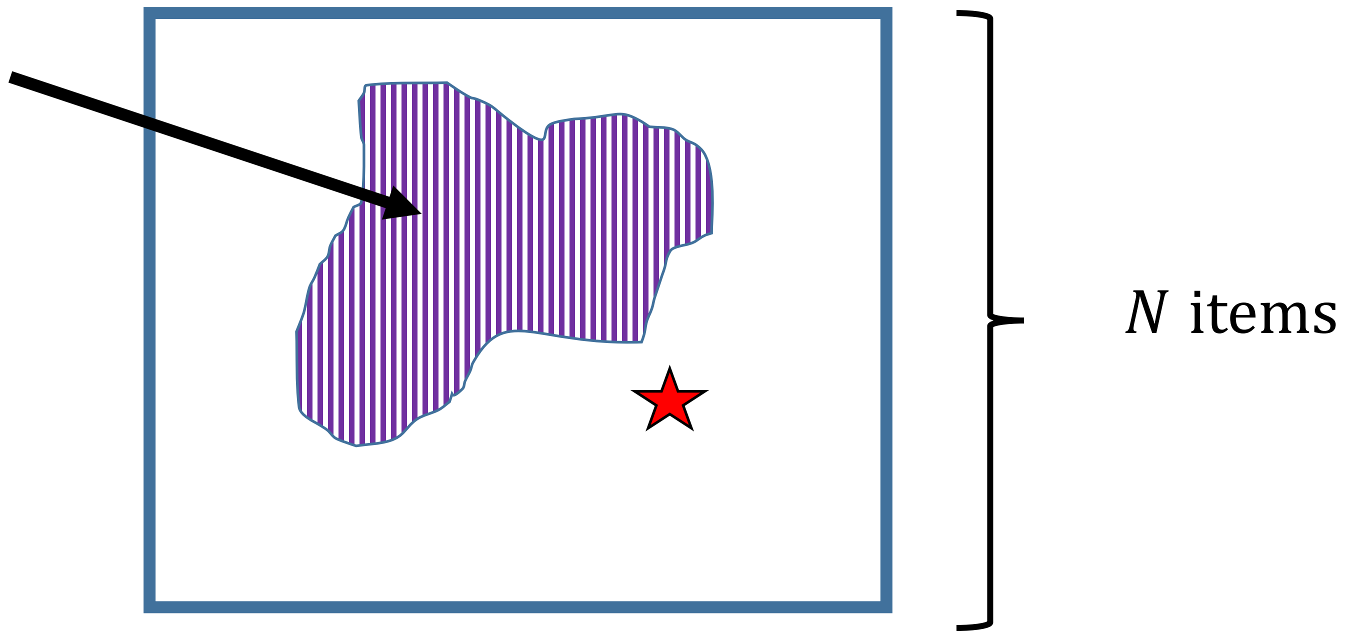


Lower Bounds for Search with Multiple Oracles

- What if only needed to use Oracle $\sqrt{M}$ times? (For contradiction.)

Idea: Even if aren't given Oracle, create it using ★ Oracle.

Choose M items at random to be marked by " Oracle"



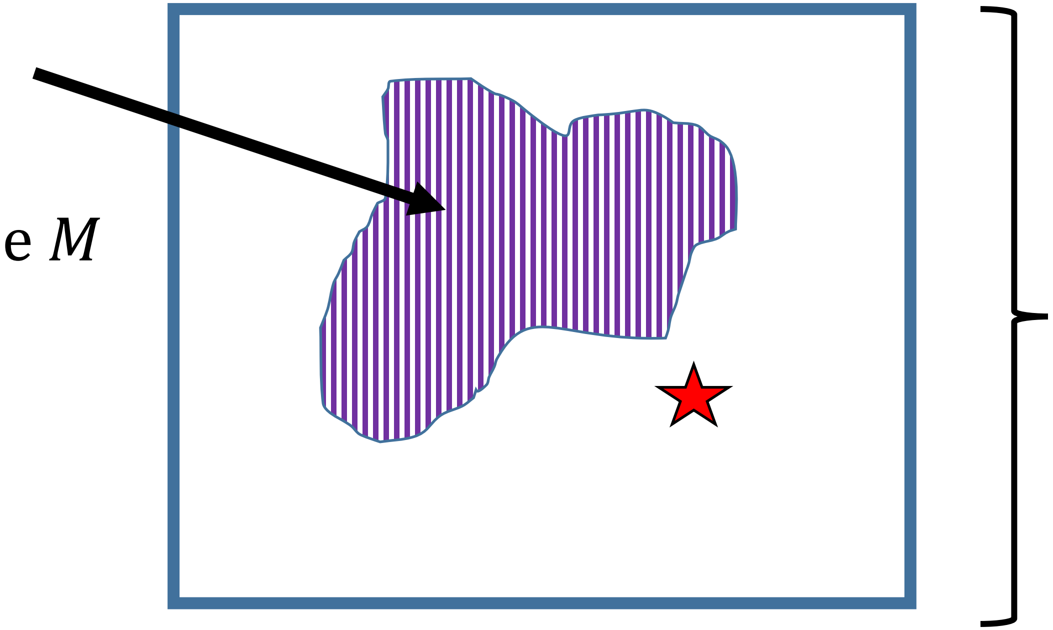
Lower Bounds for Search with Multiple Oracles

- What if only needed to use Oracle $\sqrt{M}$ times? (For contradiction.)

Idea: Even if aren't given Oracle, create it using ★ Oracle.

Choose M items at random to be marked by "Oracle"

Problem: Need starred item in the M



N items

Lower Bounds for Search with Multiple Oracles

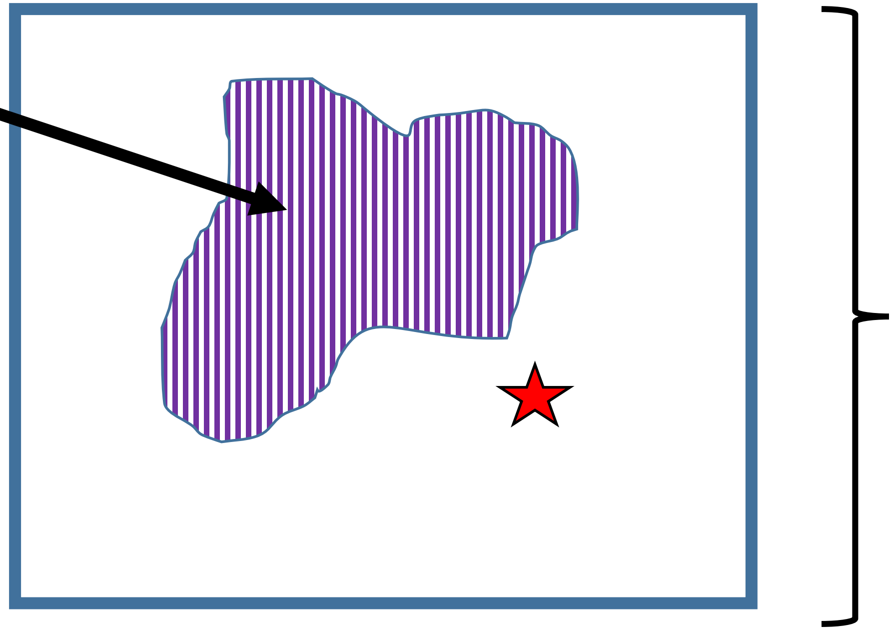
- What if only needed to use Oracle $\sqrt{M}$ times? (For contradiction.)

Idea: Even if aren't given Oracle, create it using ★ Oracle.

Choose M items at random to be marked by "Oracle"

Problem: Need starred item in the M

Solution: Use ★ Oracle to check if starred. If it is starred, mark as striped.



N items

Lower Bounds for Search with Multiple Oracles

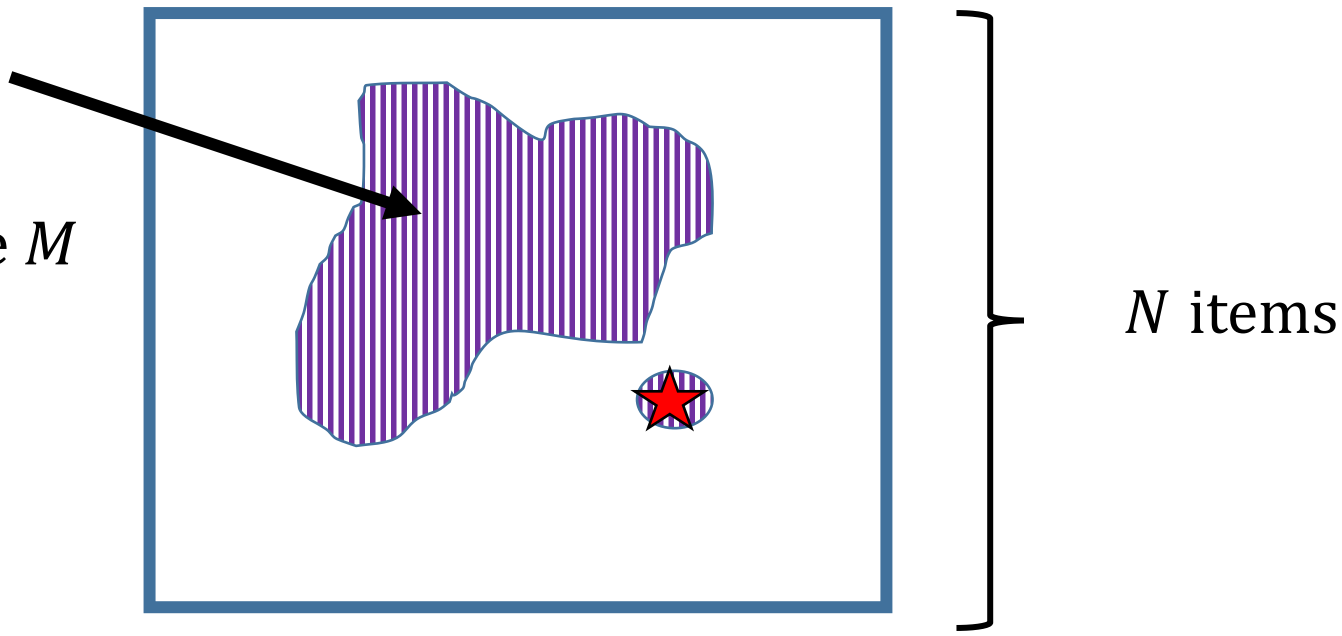
- What if only needed to use Oracle $\sqrt{M}$ times? (For contradiction.)

Idea: Even if aren't given Oracle, create it using Oracle.

Choose M items at random to be marked by " Oracle"

Problem: Need starred item in the M

Solution: Use Oracle to check if starred. If it is starred, mark as striped.



Lower Bounds for Search with Multiple Oracles

- What if only needed to use ◍ Oracle $\sqrt{M}$ times? (For contradiction.)

Idea: Even if aren't given ◍ Oracle, create it using ★ Oracle.

Can simulate ◍ Oracle using one (two) queries to ★ Oracle!

⇒ Only need to use ★ Oracle $\sim\sqrt{M}$ times to simulate ◍ Oracle.

Lower Bounds for Search with Multiple Oracles

- What if only needed to use **Oracle** $\sqrt{M}$ times? (For contradiction.)

Idea: Even if aren't given **Oracle,** create **it** using ★ **Oracle**.

Can simulate **Oracle** using one (two) queries to ★ **Oracle**!

→ Only need to use ★ **Oracle** $\sim\sqrt{M}$ times to simulate **Oracle**.

→ Previously showed need $\sim\sqrt{M}$ queries to ★ **Oracle** if have an **Oracle**.

Lower Bounds for Search with Multiple Oracles

- What if only needed to use Oracle $\sqrt{M}$ times? (For contradiction.)

Idea: Even if aren't given Oracle, create it using ★ Oracle.

Can simulate Oracle using one (two) queries to ★ Oracle!

→ Only need to use ★ Oracle $\sim\sqrt{M}$ times to simulate Oracle.

→ Previously showed need $\sim\sqrt{M}$ queries to ★ Oracle if have an Oracle.

→ Can find starred item using $\sim\sqrt{M}$ queries to ★ Oracle

Lower Bounds for Search with Multiple Oracles

- What if only needed to use ◍ **Oracle** $\sqrt{M}$ times? (For contradiction.)

Idea: Even if aren't given ◍ **Oracle,** create it using ★ **Oracle**.

Can simulate ◍ **Oracle** using one (two) queries to ★ **Oracle**!

Only need to use ★ **Oracle** $\sim\sqrt{M}$ times to simulate ◍ **Oracle**.

Previously showed need $\sim\sqrt{M}$ queries to ★ **Oracle** if have an ◍ **Oracle**.

Can find starred item using $\sim\sqrt{M}$ queries to ★ **Oracle**

TOOL

Lower Bounds for Search with Multiple Oracles

- Using this argument:

2. Always need to use either ★ **Oracle** or ◍ **Oracle** at least $\sim\sqrt{N}$ times.

Lower Bounds for Search with Multiple Oracles

1.

Always need to use ★**Oracle** at least $\sim\sqrt{M}$ times.

2.

Always need to use either ★**Oracle** or ◍ **Oracle** at least $\sim\sqrt{N}$ times.

Minimum cost:

$$c_{\parallel}\sqrt{N} + c_{\star}\sqrt{M}$$

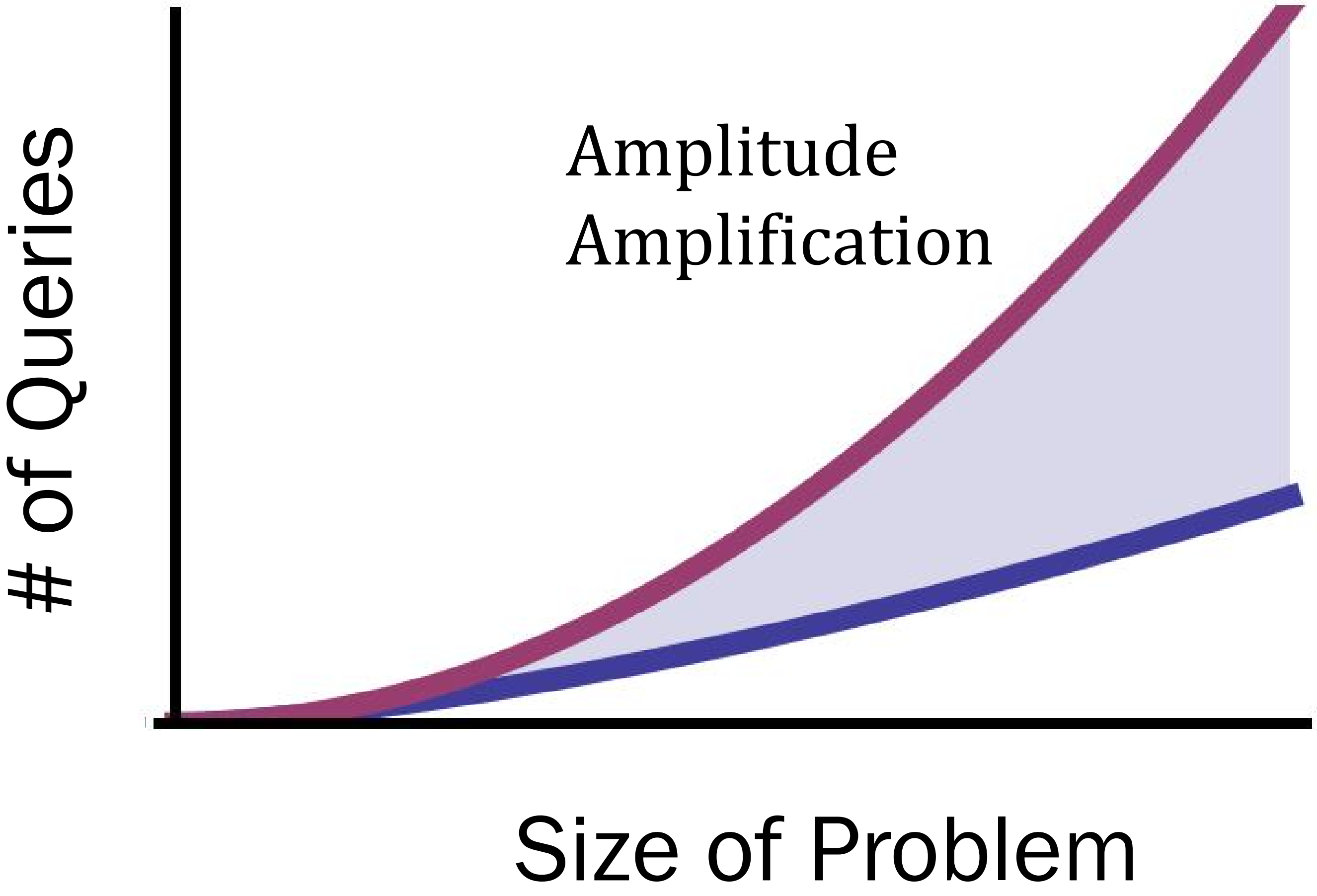
Or

$$c_{\star}\sqrt{N}$$

Algorithm for Searching with Multiple Oracles

Amplitude Amplification

Quantum Query Complexity Bounds



Almost tight! (Up to constant factors)

Simulate **Oracle** ◍
Hybrid Method
[Bennett et al. '97]

Algorithm for 3-SAT

$$F(x_1, x_2, \cdots, x_n) = (x_1 \vee \neg x_3 \vee x_7) \wedge (x_2 \vee x_3 \vee \neg x_6) \wedge (x_1 \vee x_7 \vee x_{10}) \ldots$$

$\sim$poly(n) clauses (e.g. Cn^2)

- Guess a satisfying assignment. Test if all clauses are satisfied **EXPENSIVE**
  - Need to test $\sim 2^n$ possible inputs. With quantum computer can do in $\sqrt{2^n}$ steps
- Guess a satisfying assignment. Test if $\sim \log(n)$ clauses are satisfied **CHEAP**
  Defines a subset of possible solutions, including the true satisfying assignment, if it exists

What is *M*?

Directions for Future Work

- Create tight bounds for searching with multiple oracles
  - Adversary Bound/Span programs
  - Geometric picture

- Can we create a general framework for understanding oracles with costs, in the way that the adversary bound is a framework for understanding standard oracle problems
- Many quantum oracle problems – does it make sense to add multiple oracles to these problems?

Classical Algorithm

1. Choose item at random and test if striped using Oracle 2
2. If it is striped, test if starred using Oracle 1

Worst case cost:

$$c_1(M-1) + c_2 N$$