Problems with Multiple Oracles

Shelby Kimmel
Center for Theoretical Physics, MIT

Coogee 2014

Example: 3-SAT

Is there an $x \in \{0,1\}^n$ that satisfies $f(x) = 1$, where

$$f(x) = (x_1 \vee \neg x_3 \vee x_7) \wedge (x_2 \vee x_3 \vee \neg x_6) \wedge (x_1 \vee x_7 \vee x_{10}) \wedge \cdots.$$

Clauses ($\sim$poly(n) total)

Example: 3-SAT

Is there an $x \in \{0,1\}^n$ that satisfies $f(x) = 1$, where

$$f(x) = (x_1 \vee \neg x_3 \vee x_7) \wedge (x_2 \vee x_3 \vee \neg x_6) \wedge (x_1 \vee x_7 \vee x_{10}) \wedge \cdots .$$

Clauses ($\sim$poly(n) total)

- Given x, can test if all clauses are satisfied.
  - $\sim 2^n$ possible x. With quantum computer need $\sim\sqrt{2^n}$ steps

Example: 3-SAT

Is there an $x \in \{0,1\}^n$ that satisfies $f(x) = 1$, where

$$f(x) = (x_1 \vee \neg x_3 \vee x_7) \wedge (x_2 \vee x_3 \vee \neg x_6) \wedge (x_1 \vee x_7 \vee x_{10}) \wedge \cdots .$$

Clauses ($\sim$poly(n) total)

- Given x, can test if all clauses are satisfied.
  - $\sim 2^n$ possible x. With quantum computer need $\sim\sqrt{2^n}$ steps
- Given x, can test if some set of $\sim\log(n)$ of the clauses are satisfied.
  - Identifies subset of possible x that includes satisfying x, if it exists

Example: 3-SAT

Is there an $x \in \{0,1\}^n$ that satisfies $f(x) = 1$, where

$$f(x) = (x_1 \vee \neg x_3 \vee x_7) \wedge (x_2 \vee x_3 \vee \neg x_6) \wedge (x_1 \vee x_7 \vee x_{10}) \wedge \cdots.$$

Clauses ($\sim$poly(n) total)

- Given x, can test if all clauses are satisfied. **EXPENSIVE**
  - $\sim 2^n$ possible x. With quantum computer need $\sim\sqrt{2^n}$ steps
- Given x, can test if some set of $\sim\log(n)$ of the clauses are satisfied. **CHEAP**
  - Identifies subset of possible x that includes satisfying x, if it exists

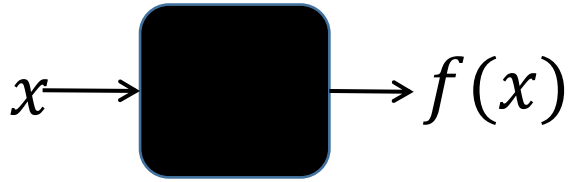
Outline

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

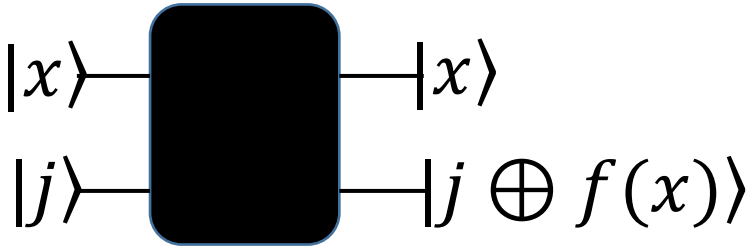
Standard Oracle Model

Goal: Determine a property of the function $f(x)$, given an oracle for f

Classical Oracle



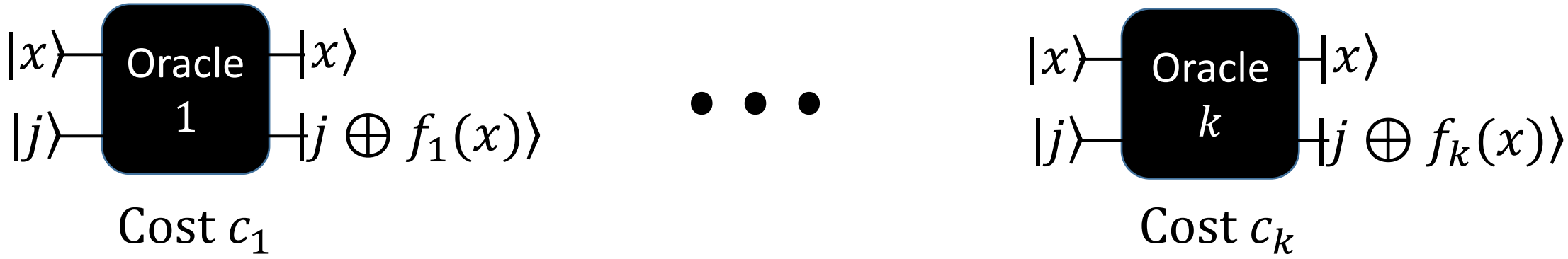
Quantum Oracle



Only care about # of oracle uses (queries)

Multiple Oracles with Costs Model

Goal: Determine a property of $f(x)$, given set of oracles associated with functions $\{f_1, \cdots, f_k\}$ which each have some information related to f



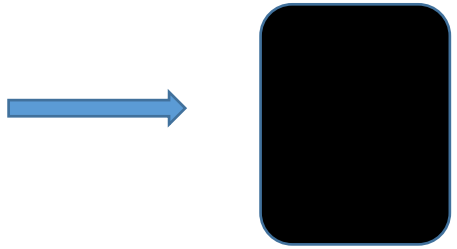
Care about total cost $= \sum_{i=1}^{k} q_i c_i$ where q_i is the # of times Oracle i is used

Utility of Multiple Oracles Model

Step away from “black boxes,” while retaining tools of oracles

- In the real world oracles are not “black boxes” and we often have extra information about the function f

$$f(x) = ((x_1 \wedge x_3) \vee x_7) \wedge (x_2 \wedge (x_3 \vee x_6)) \wedge (x_1 \vee x_7 \vee x_{10}) \ldots.$$



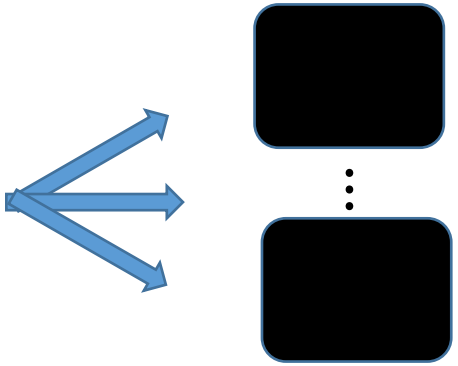
- In the real world, oracles take time to implement
- Can apply oracle tool box: algorithms, lower bounding techniques, etc

Utility of Multiple Oracles Model

Step away from “black boxes,” while retaining tools of oracles

- In the real world oracles are not “black boxes” and we often have extra information about the function f

$$f(x) = ((x_1 \wedge x_3) \vee x_7) \wedge (x_2 \wedge (x_3 \vee x_6)) \wedge (x_1 \vee x_7 \vee x_{10}) \ldots.$$



- In the real world, oracles take time to implement
- Can apply oracle tool box: algorithms, lower bounding techniques, etc

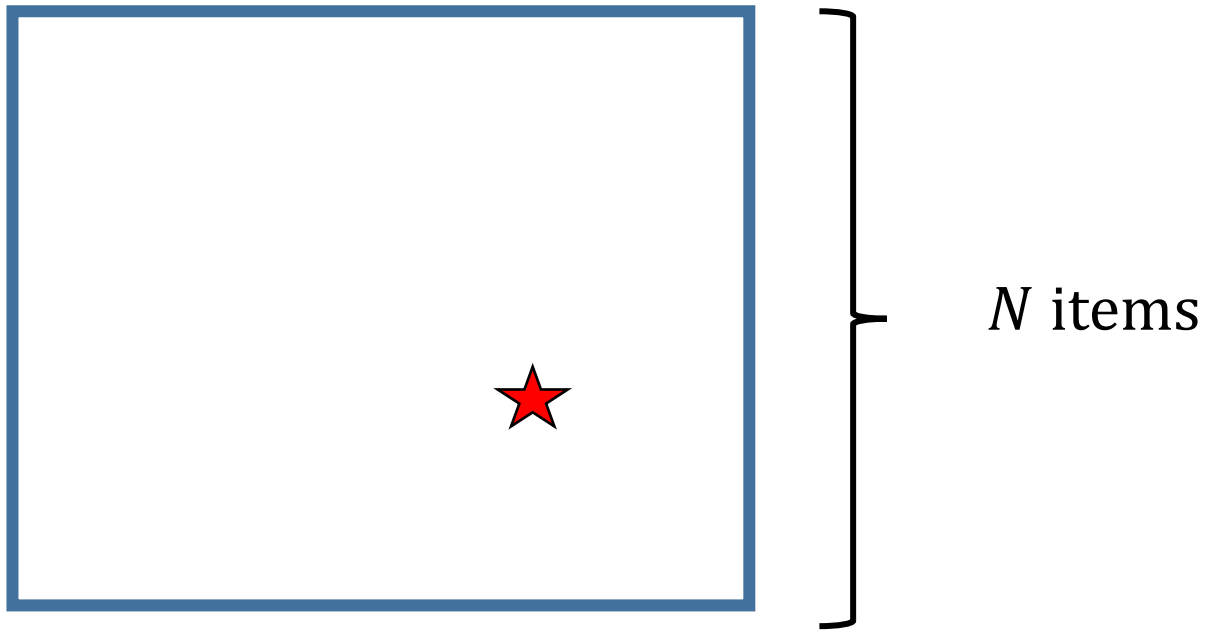
Related Work

- 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: One oracle, but given some additional information about the solution.
  - 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.
  - Need certain structure, No cost, No lower bounds,

Searching with an Oracle

Can ask oracle, "Is the i^{th} item the starred item?"

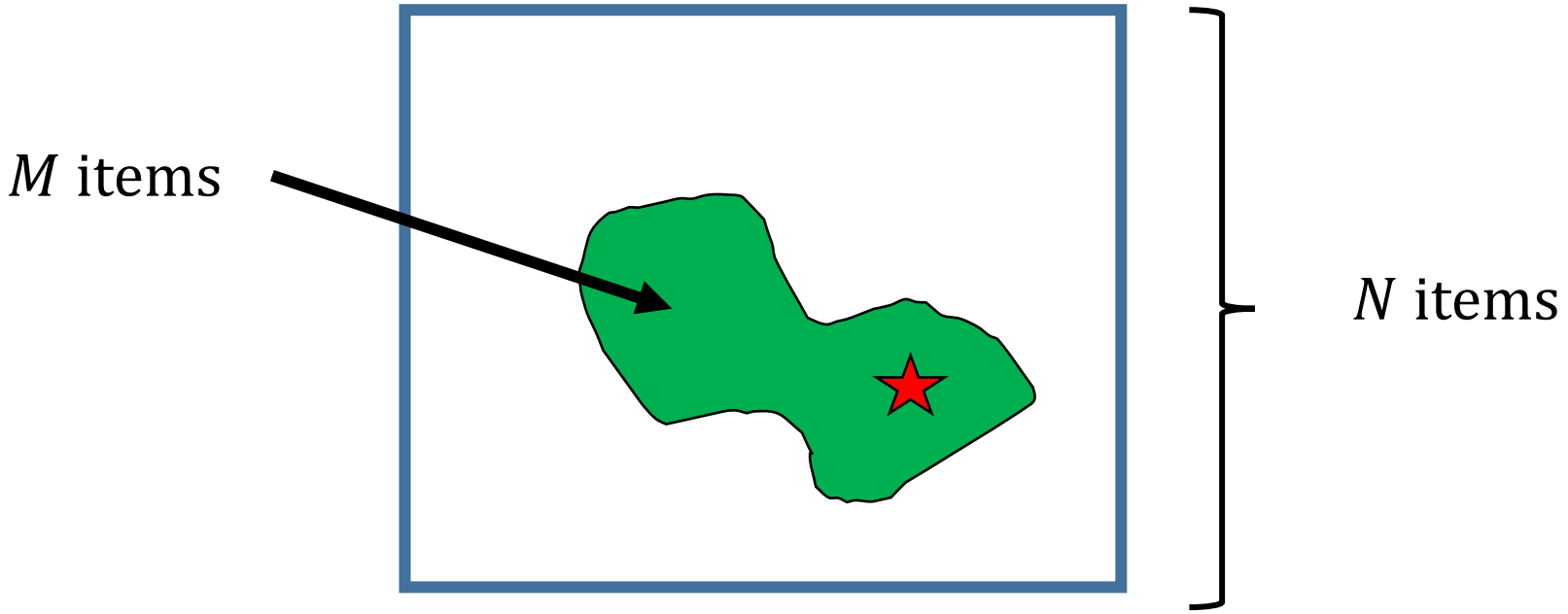
- Classically, need $\Theta(N)$ queries to oracle
- Quantumly, need $\Theta(\sqrt{N})$ queries to oracle [Grover '97, Bennett et al. '97, Zalka '99]



Searching with Multiple Oracles

Can ask **Oracle 1**, "Is the i^{th} item starred?"
Can ask **Oracle 2**, "Is the i^{th} item green?"

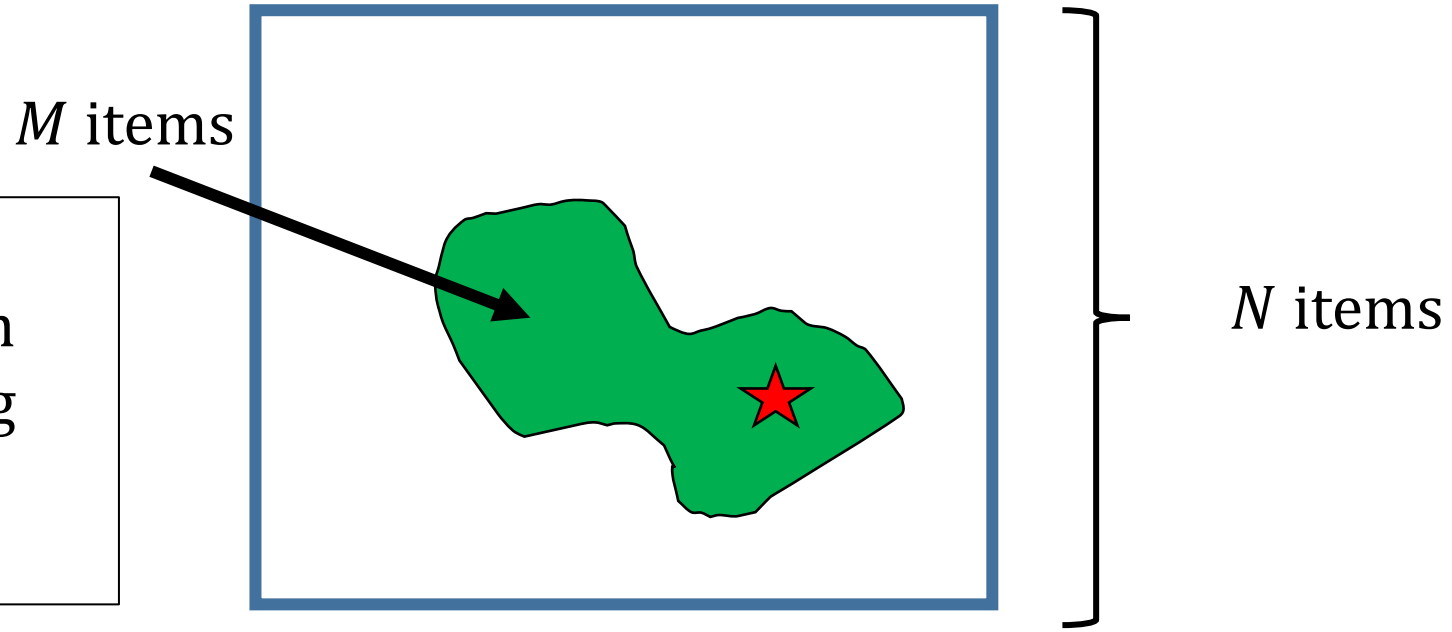
Promised: The starred item is also green



Searching with Multiple Oracles

Can ask **Oracle 1**, "Is the i^{th} item starred?"
Can ask **Oracle 2**, "Is the i^{th} item green?"

Promised: The starred item is also green

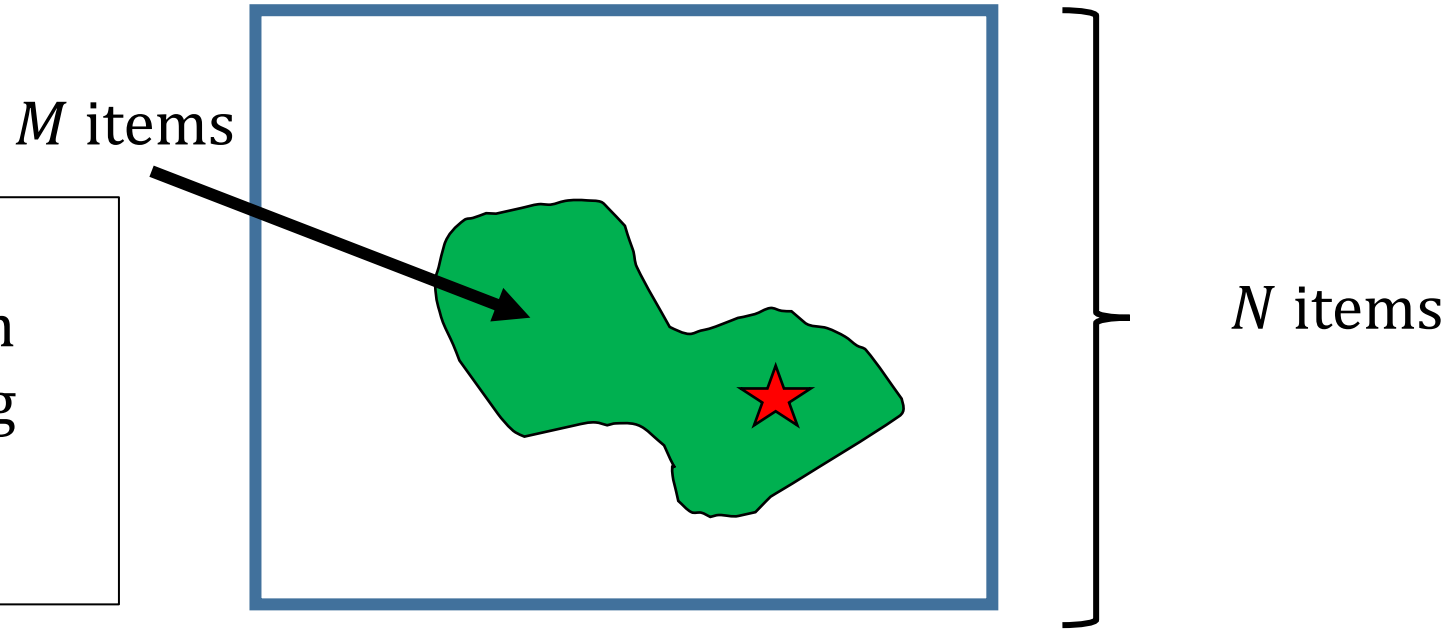


Can you find the starred item with fewer steps using **Oracle 2**?
…

Searching with Multiple Oracles

Can ask **Oracle 1**, "Is the i^{th} item starred?"
Can ask **Oracle 2**, "Is the i^{th} item green?"

Promised: The starred item is also green



Can you find the starred item with fewer steps using **Oracle 2**?
… **NO**

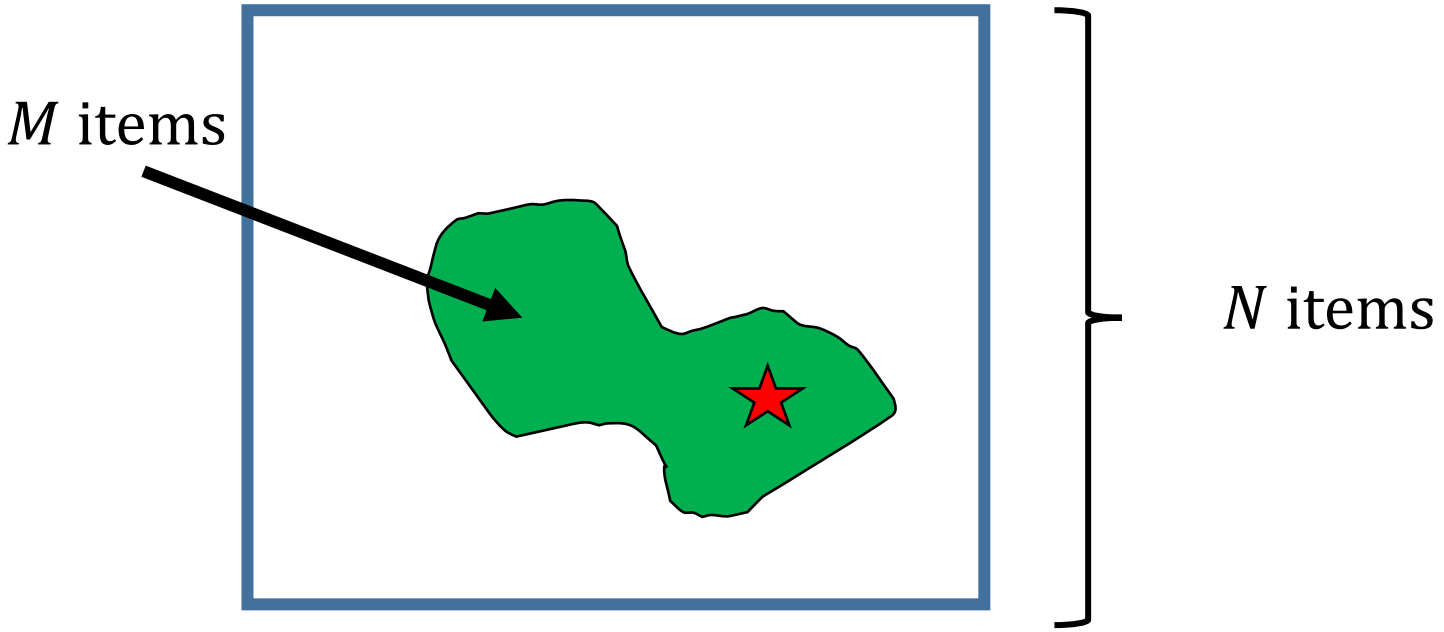
Searching with Multiple Oracles

Can ask **Oracle 1**, "Is the i^{th} item starred?" **with cost c_1**

Can ask **Oracle 2**, "Is the i^{th} item green?" **with cost c_2**

$c_1 > c_2$

Promised: The starred item is also green



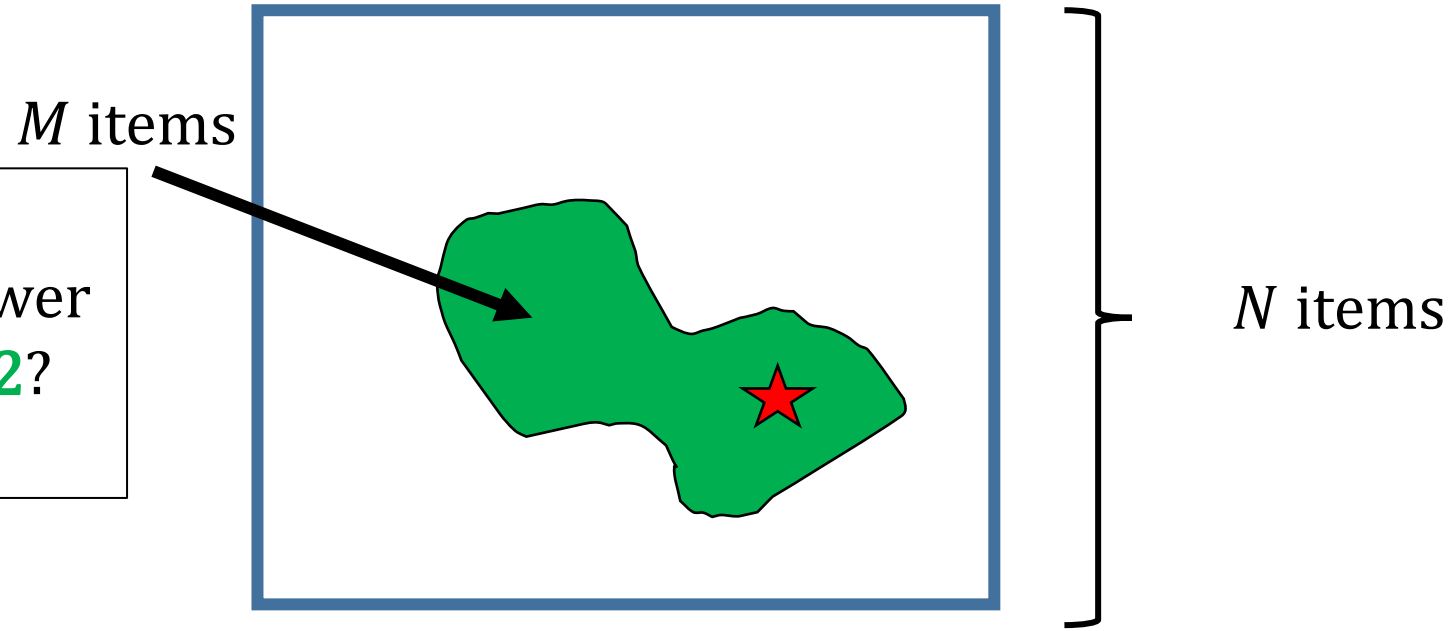
Searching with Multiple Oracles

Can ask **Oracle 1**, "Is the i^{th} item starred?" **with cost c_1**

Can ask **Oracle 2**, "Is the i^{th} item green?" **with cost c_2**

$c_1 > c_2$

Promised: The starred item is also green



Can you find the starred item at lower cost using **Oracle 2**?

…

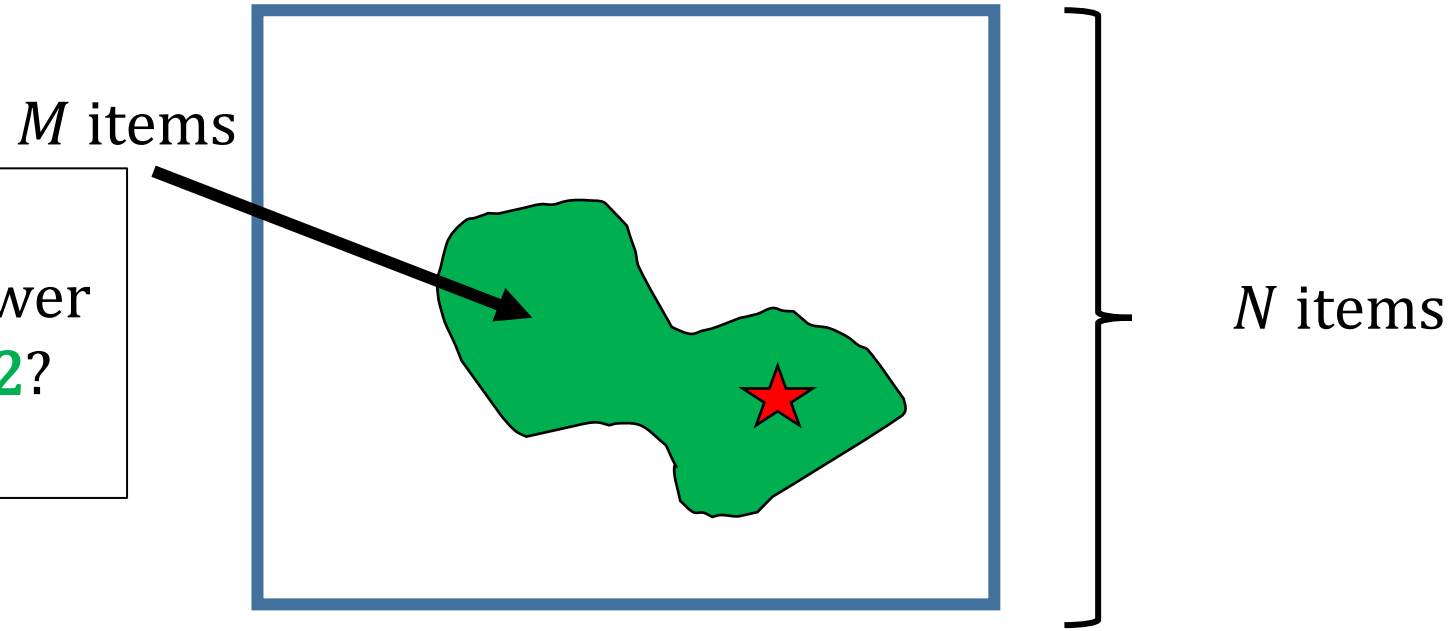
Searching with Multiple Oracles

Can ask **Oracle 1**, "Is the i^{th} item starred?" **with cost c_1**

Can ask **Oracle 2**, "Is the i^{th} item green?" **with cost c_2**

$c_1 > c_2$

Promised: The starred item is also green



Can you find the starred item at lower cost using **Oracle 2**?
… **YES**

Searching with Multiple Oracles

Can ask **Oracle 1**, "Is the i^{th} item starred?" **with cost c_1**

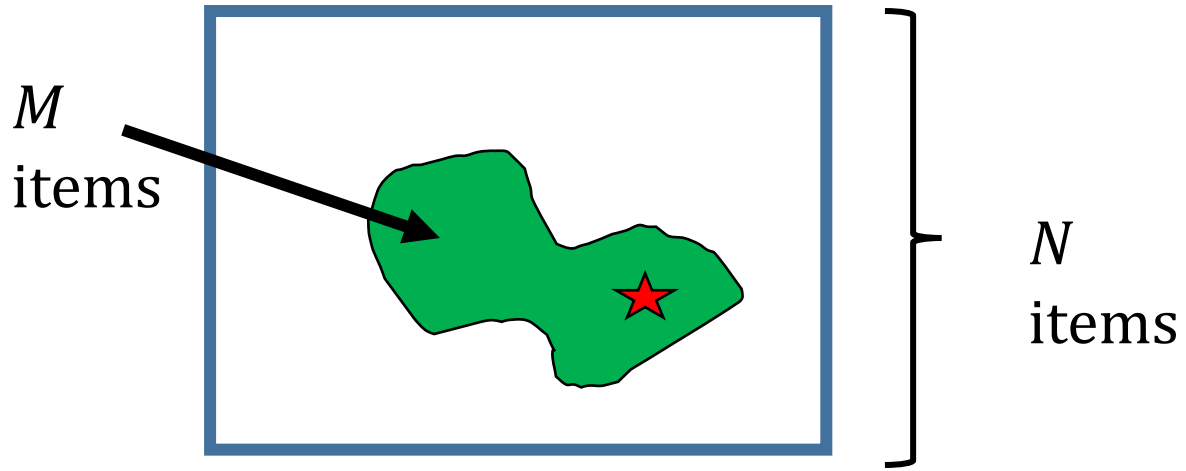
Can ask **Oracle 2**, "Is the i^{th} item green?" **with cost c_2**

$c_1 > c_2$

Promised: The starred item is also green

Classical $\sim \min\{c_1 N, c_2 N + c_1 M\}$

Quantum $\sim \min\{c_1\sqrt{N}, c_2\sqrt{N} + c_1\sqrt{M}\}$



Quantum Algorithm for Searching with Multiple Oracles

Amplitude amplification:

- Suppose have oracle $\mathcal{O}$ s.t.

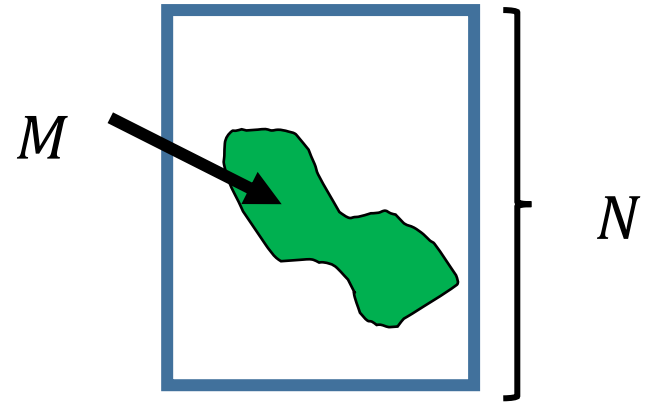
$$\mathcal{O}|\psi_{good}\rangle = |\psi_{good}\rangle|1\rangle$$
$$\mathcal{O}|\neg\psi_{good}\rangle = |\neg\psi_{good}\rangle|0\rangle$$

- Given reversible algorithm $\mathcal{A}$

$$\mathcal{A}|\psi_{initial}\rangle = \sqrt{p}|\psi_{good}\rangle + \sqrt{1-p}|\neg\psi_{good}\rangle$$

- Can create Grover-like algorithm that succeeds with constant probability using
  - ~$2\sqrt{p^{-1}}$ applications of $\mathcal{A}$ / $\mathcal{A}^{-1}$,
  - $\sim\sqrt{p^{-1}}$ applications of $\mathcal{O}$

Quantum Algorithm for Searching with Multiple Oracles



- Have **Oracle 1** ($\mathcal{O}_1$) s.t.

$$\mathcal{O}_1|i\rangle = |i\rangle|1\rangle \text{ if i is starred}$$
$$\mathcal{O}_1|i\rangle = |i\rangle|0\rangle \text{ if } i \text{ is not starred}$$

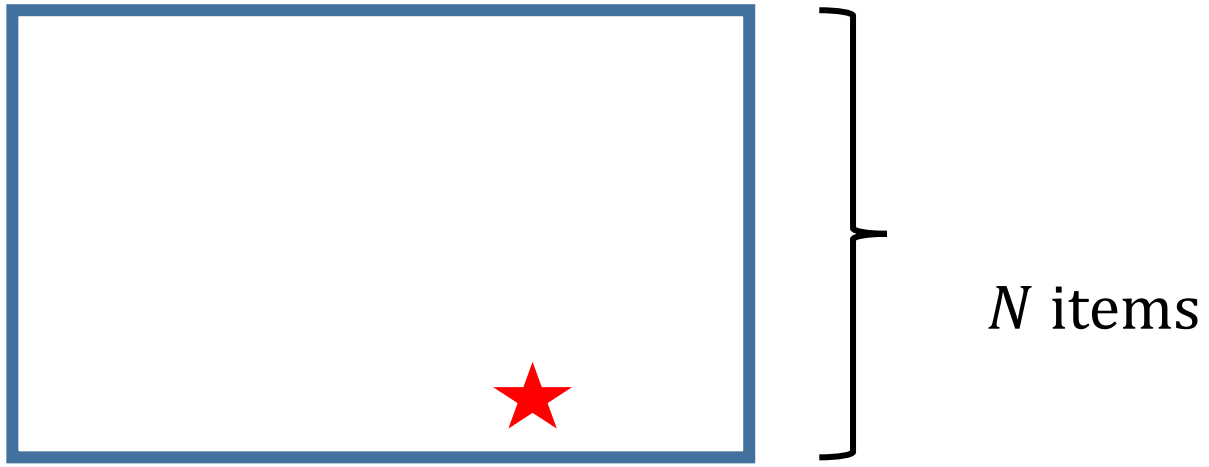
- Run Grover's algorithm with $\sqrt{N/M}$ queries **Oracle 2** to create algorithm $\mathcal{A}$:

$$\mathcal{A}|\psi_{initial}\rangle = \frac{1}{\sqrt{M}} \sum_{i\ green} |i\rangle = \frac{1}{\sqrt{M}}|i_{starred}\rangle + \frac{1}{\sqrt{M}} \sum_{\substack{i\ green \\ not\ starred}} |i\rangle$$

- Using amplitude amplification can create algorithm that finds starred item
  - $\sim 2\sqrt{M}$ applications of $\mathcal{A}$ / $\mathcal{A}^{-1} \Longrightarrow \sqrt{M} \times \sqrt{N/M} = \sim\sqrt{N}$ application of **Oracle 2**
  - $\sim\sqrt{M}$ applications of **Oracle 1** $\mathcal{O}_1$,

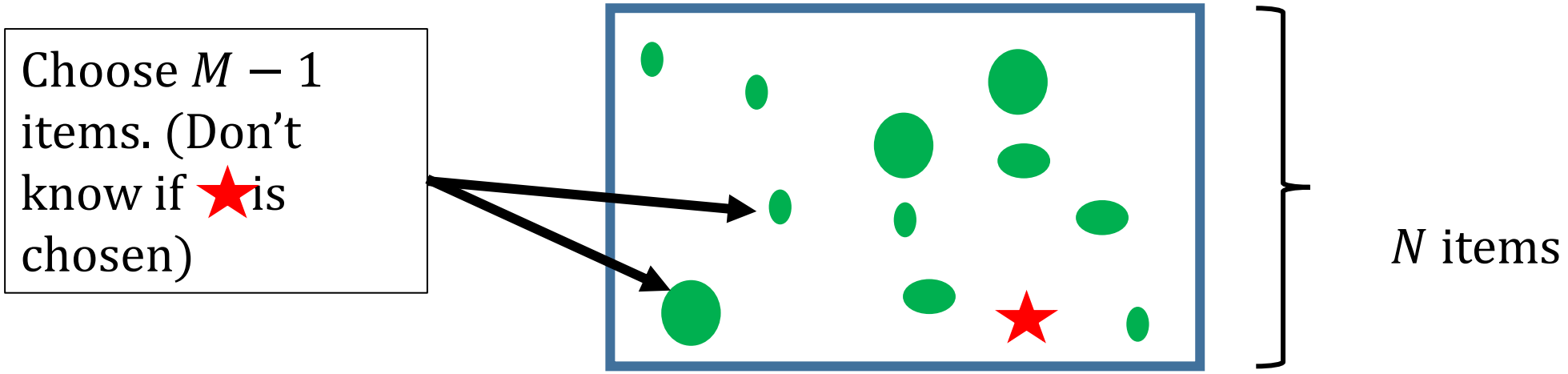
Lower Bounds for Search with Multiple Oracles

- Using just **Oracle 1**, need $\Omega(\sqrt{N})$ queries [Bennet et al. '97]
- Using just **Oracle 1**, can create an **Oracle 2**



Lower Bounds for Search with Multiple Oracles

- Using just **Oracle 1**, need $\Omega(\sqrt{N})$ queries [Bennet et al. '97]
- Using just **Oracle 1**, can create an **Oracle 2**



If algorithm uses **Oracle 1** q_1 times, **Oracle 2** q_1 times, we must have: $q_1 + q_2 = \Omega(\sqrt{N})$

Lower Bounds for Search with Multiple Oracles

- Even if have perfect knowledge of **Oracle 2**, only narrows down to M possible items.
- Need $\Omega(\sqrt{M})$ queries to **Oracle 1** [Bennet et al. '97]

Asymptotic Optimality of Algorithm

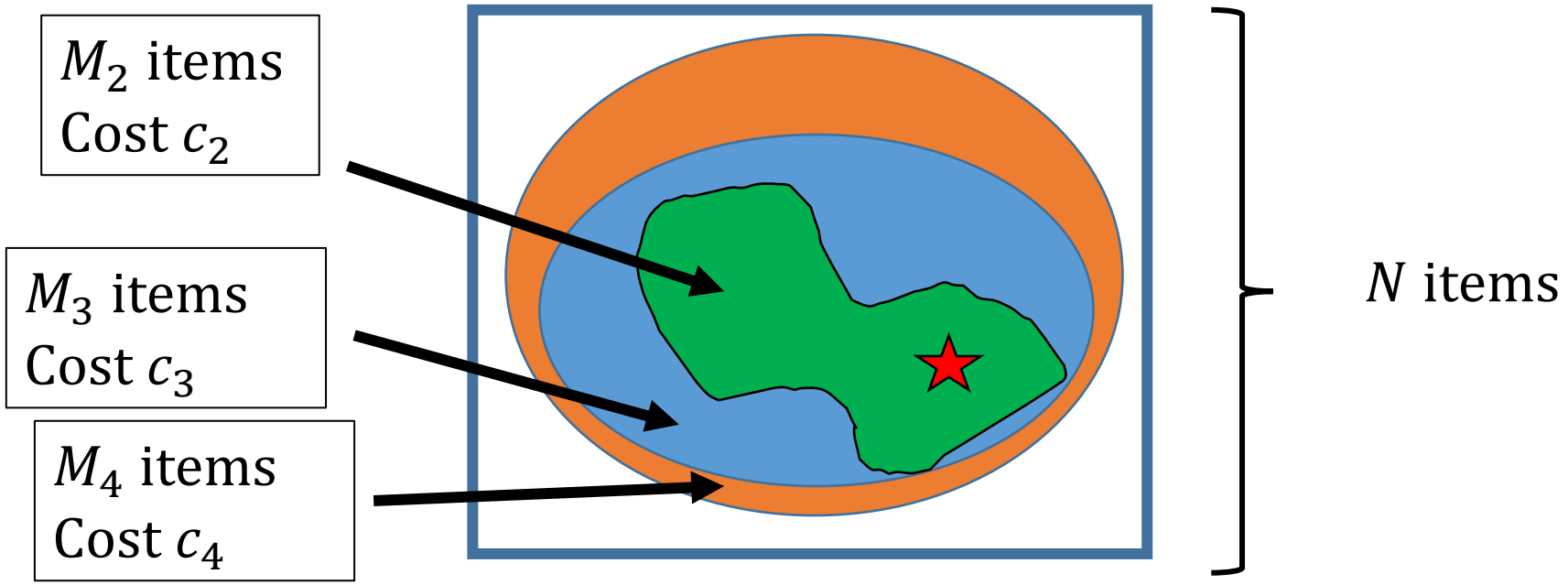
- Lower bounds
  - ❑ Need $\Omega(\sqrt{N})$ queries total
  - ❑ Need $\Omega(\sqrt{M})$ queries to **Oracle 1**
- Upper bound
  - ❑ Uses $O(\sqrt{N})$ queries total: $O(\sqrt{N})$ to **Oracle 2** and $O(\sqrt{M})$ to **Oracle 1**
  - ❑ Uses $O(\sqrt{M})$ queries to **Oracle 1**

Asymptotic Optimality of Algorithm

- Lower bounds
  - ❑ Need $\Omega(\sqrt{N})$ queries total
  - ❑ Need $\Omega(\sqrt{M})$ queries to **Oracle 1**
- Upper bound
  - ❑ Uses $O(\sqrt{N})$ queries total: $O(\sqrt{N})$ to **Oracle 2** and $O(\sqrt{M})$ to **Oracle 1**
  - ❑ Uses $O(\sqrt{M})$ queries to **Oracle 1**

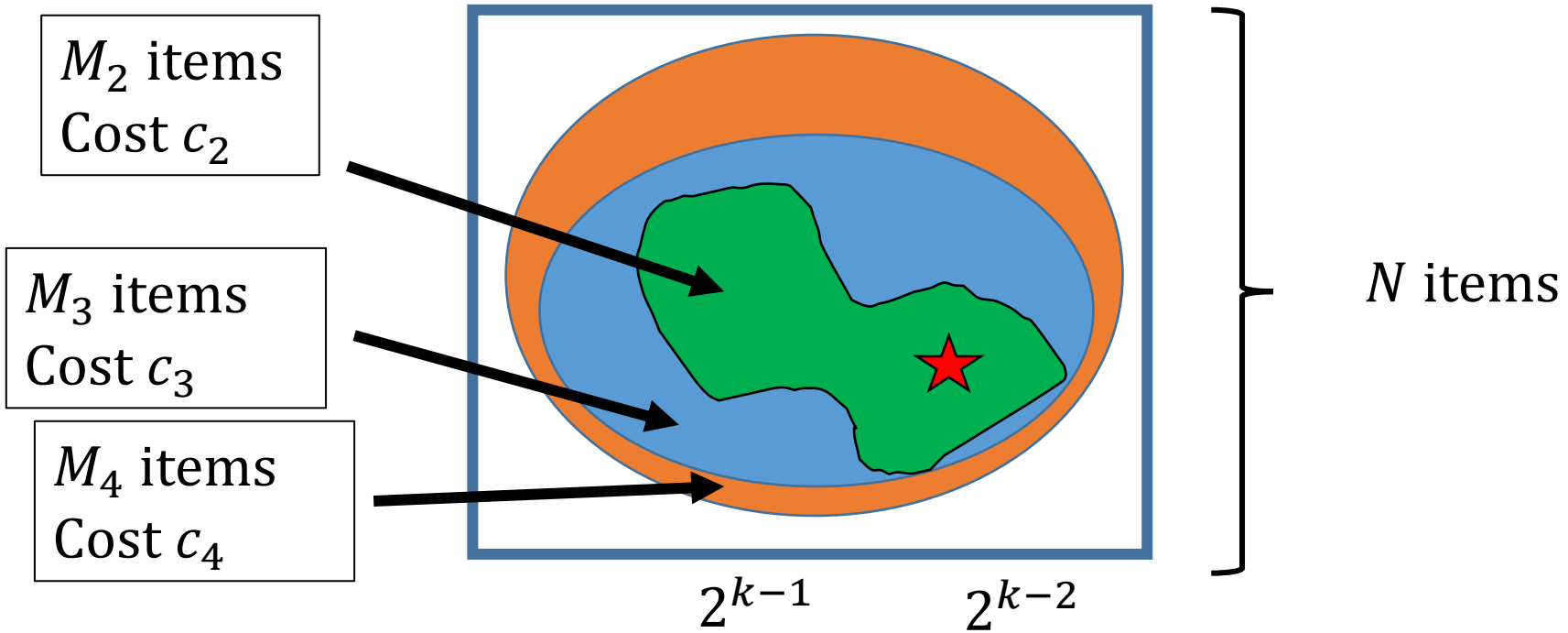
Slightly unsatisfying – is amplitude amplification the best we can do? The constants in front of the asymptotic expressions matter if costs are constant

Multiply Nested Search



Guess for Algorithm Cost: $c_k\sqrt{N} + c_{k-1}\sqrt{M_k} + \cdots + c_1\sqrt{M_2}$

Multiply Nested Search



Guess for Algorithm Cost: $c_k\sqrt{N} + c_{k-1}\sqrt{M_k} + \cdots + c_1\sqrt{M_2}$

Lower bound: $\sqrt{N}$ Total oracle uses

Classical Algorithm

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

Worst case cost:

$$c_1 M + c_2 N$$

Or can just ignore **Oracle 2** :

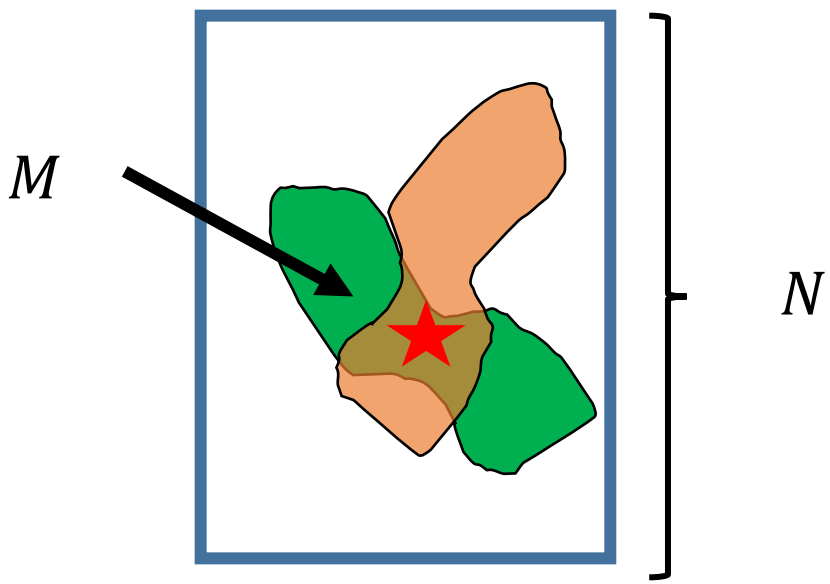
$$c_1 N$$

Compare to quantum

$$\min\{c_1\sqrt{N}, c_1\sqrt{M} + c_2\sqrt{N}\}$$

Directions for Future Work

- Create tight bounds for searching with multiple oracles
  - Adversary Bound/Span programs
  - Polynomial Method
  - Geometric picture
- General framework for understanding oracles with costs?
- Many quantum oracle problems exist– can you add extra oracles to these problems?
- Can we get a speed up for problems like 3-SAT using these techniques?
  - Current time $\sim \text{poly}(n)\ \sqrt{2^n}$. Can you achieve $\sqrt{2^n}$?


M
N