Grover's Search Algorithm

Learning Goals

2nd most famous, "Grover's Algorithm" [QCS]

- Practice analyzing a new q. algorithm w/ geometric technique
- Learn an important q. algorithmic subroutine

Search Problem

Input:

Output:

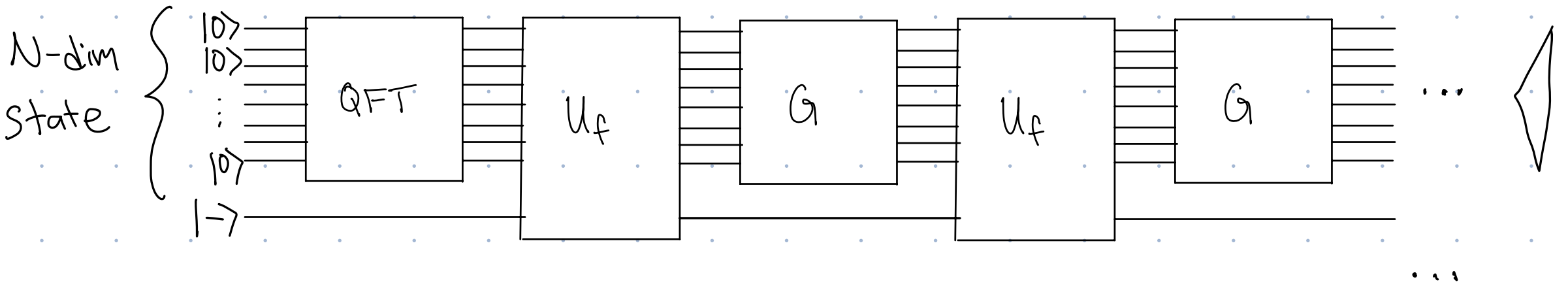
What is the classical query complexity (deterministic)?

A) $O(1)$ B) $O(\log N)$ C) $O(N)$ D) $O(2^N)$

What is the classical query complexity (probabilistic)?

A) $O(1)$ B) $O(\log N)$ C) $O(N)$ D) $O(2^N)$

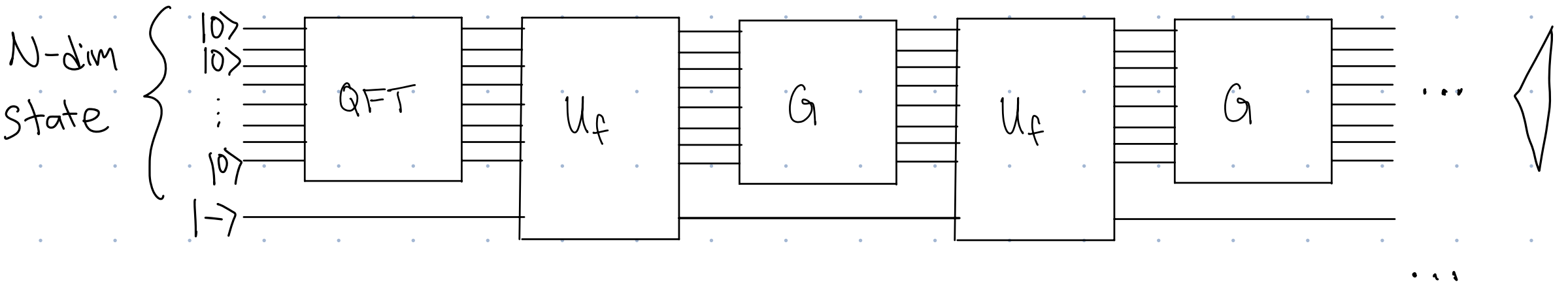
Grover's Algorithm (Quantum Search Alg)



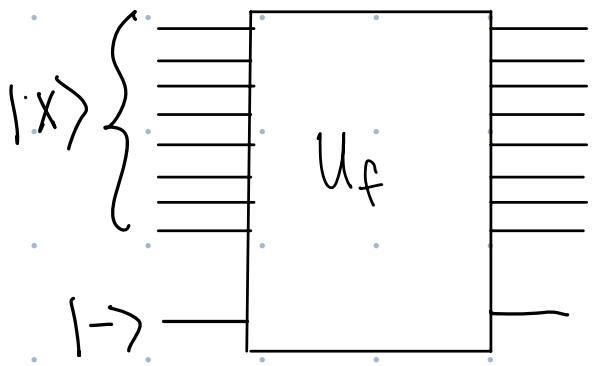
How many qubits are needed to search N possible inputs to f?

A) $\log_2 N$ B) $\log_2 N + 1$ C) N D) $N + 1$

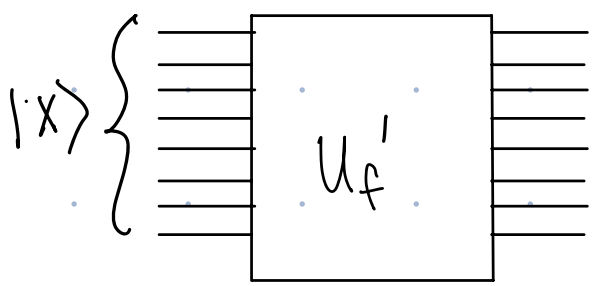
Grover's Algorithm (Quantum Search Alg)



U_f:

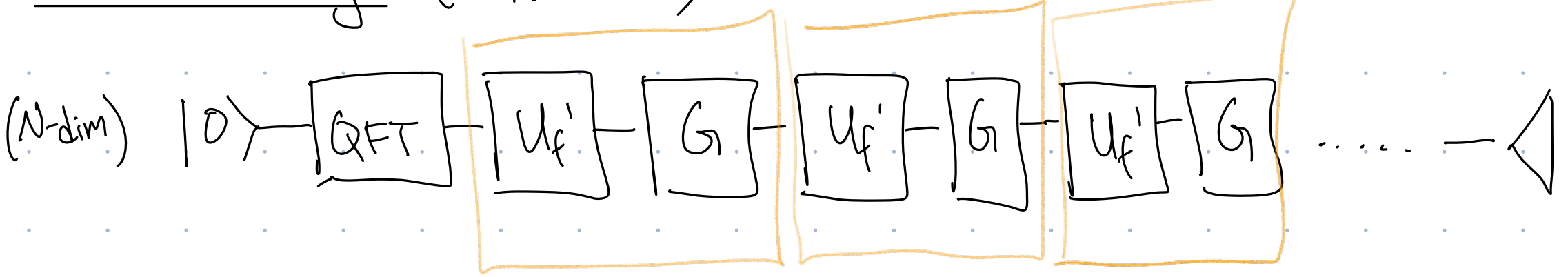






G_i (Grover Diffusion Operator

Grovers Alg: (Effective)

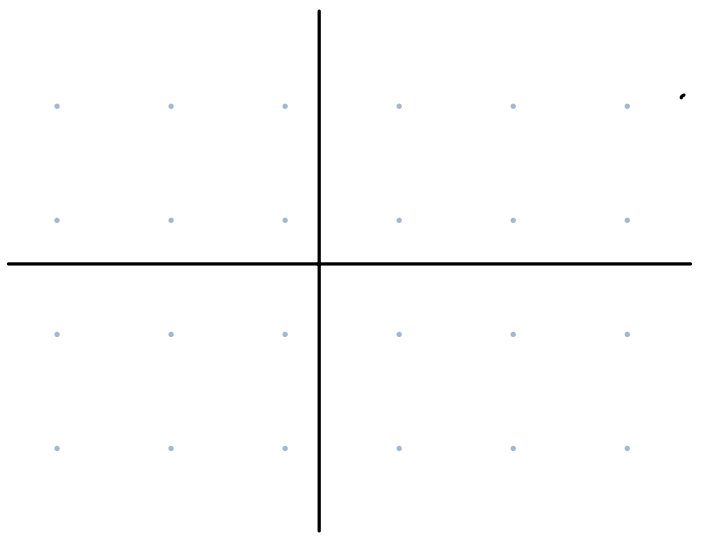


While the state is N-Dimensional, throughout the alg, state only stays in 2 of those N dimensions.

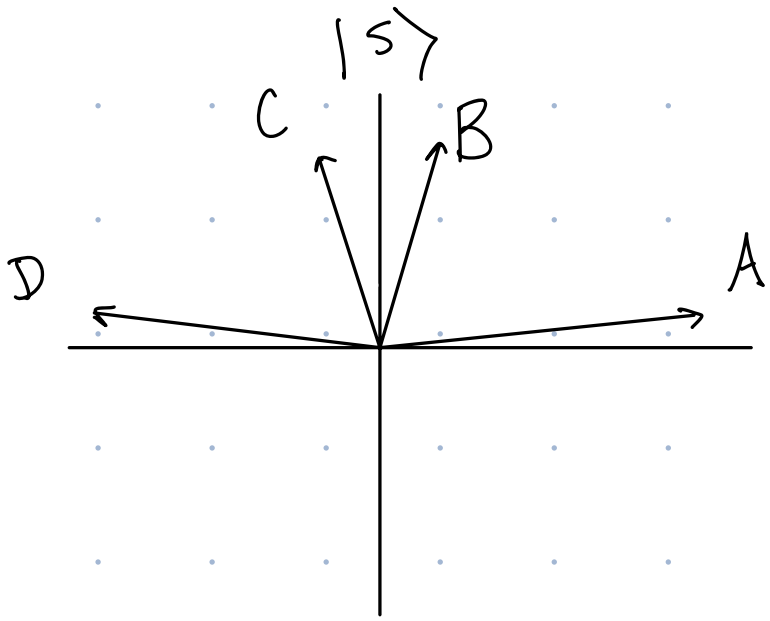
In other words, we can describe the state as a superposition of 2 states instead of N.

In particular:

We can express $|\psi_i\rangle = a_i|\beta\rangle + b_i|s\rangle$ as a vector in 2D:

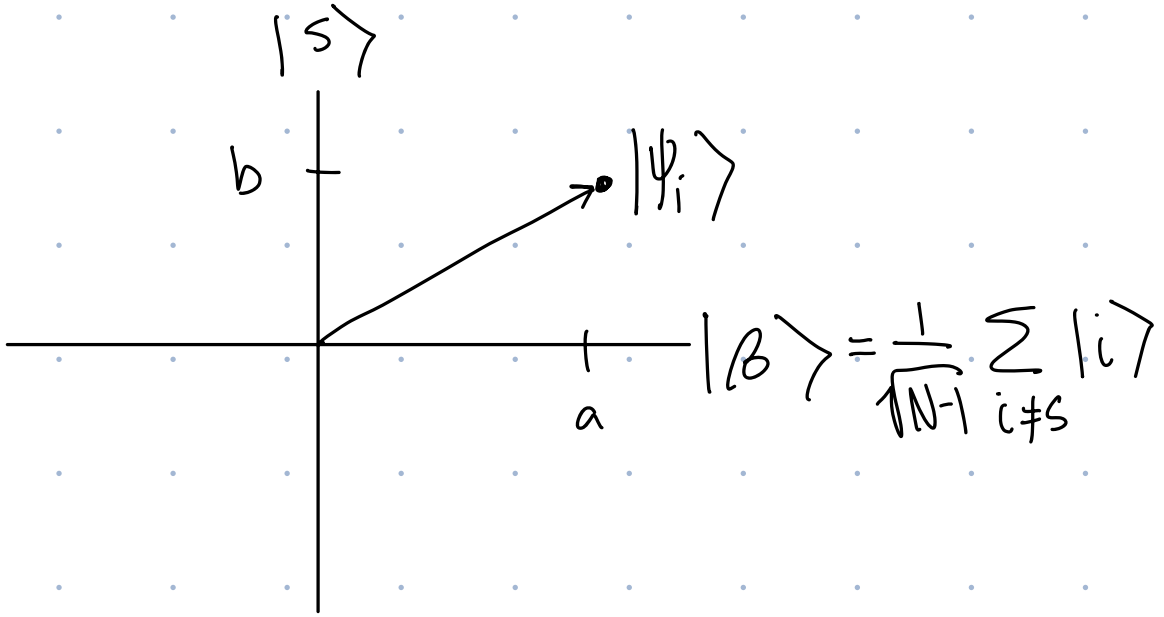


Where is $|\alpha\rangle = \frac{1}{\sqrt{N}} \sum_i |i\rangle$? (assume N is big)

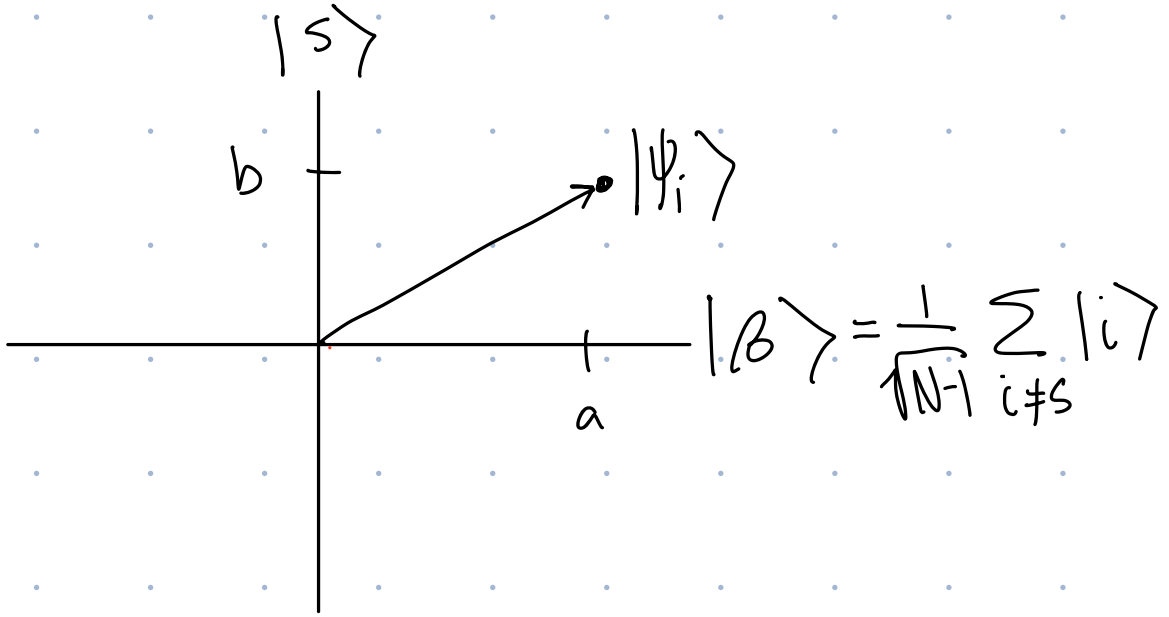


$$|B\rangle = \frac{1}{\sqrt{N-1}} \sum_{i \neq s} |i\rangle$$

Effect of $U_f' = I - 2|s\rangle\langle s|$

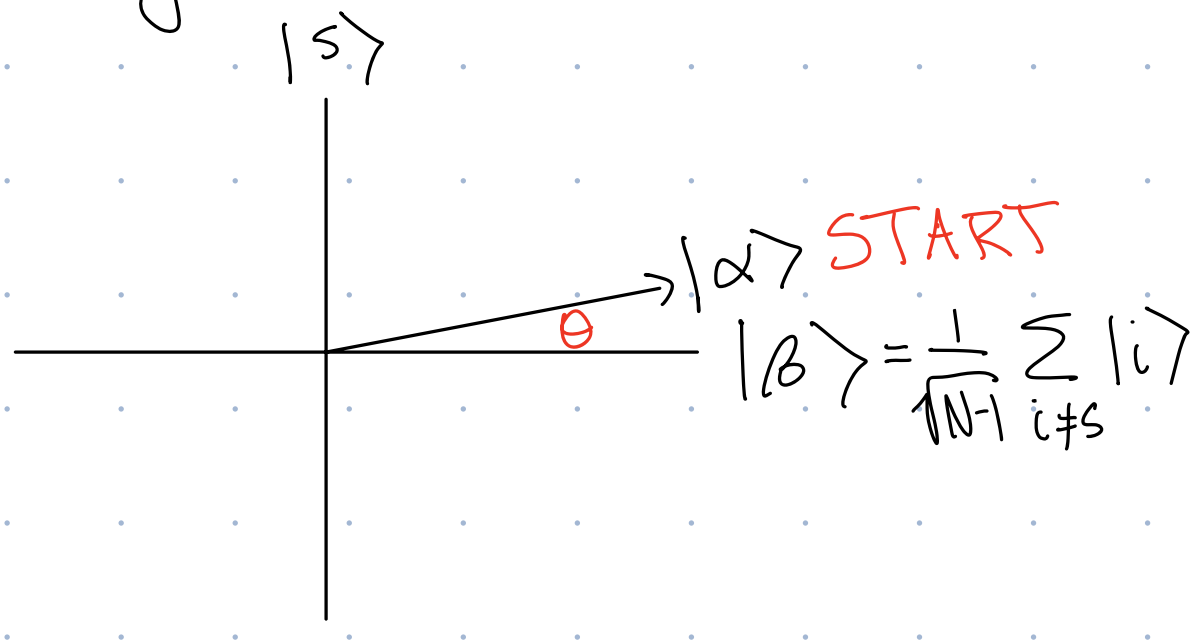


Effect of $G = -I + 2|\alpha\rangle\langle\alpha|$



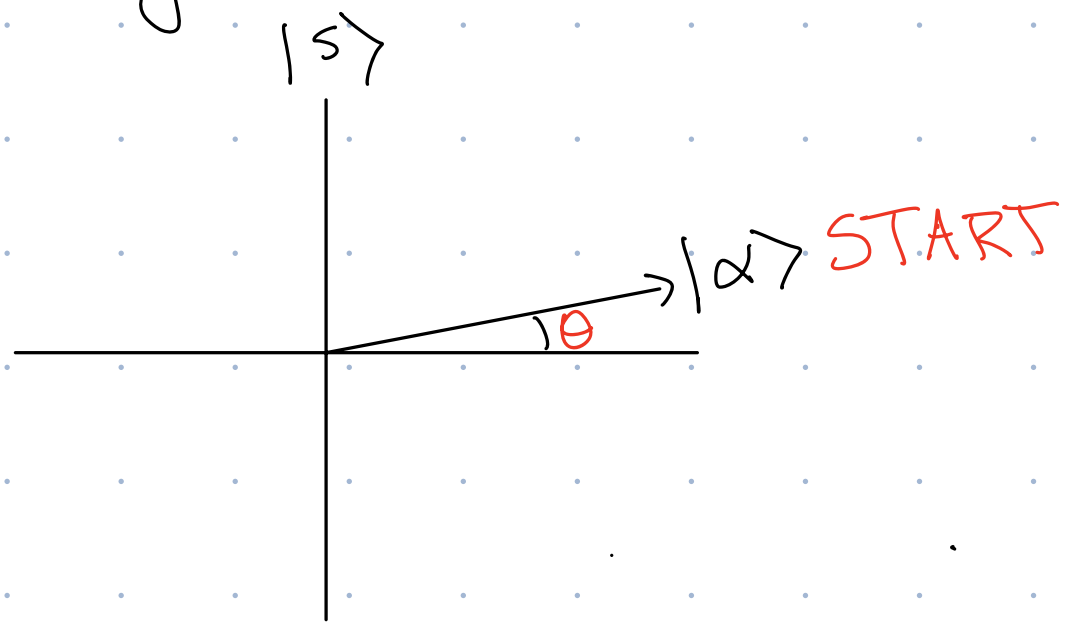
How many iterations before state becomes $|s\rangle$?

$(U_f', G, U_f', G, \ldots)$



$\tan^{-1}(x) \approx x$ for $x \ll 1$

How many iterations before state becomes $|s\rangle$?



Application: Speed Up Classical Probabilistic Algorithms:

- Probabilistic alg makes choices based on a sequence of coin flips
- Can create a classical function f that takes in any coinflip sequence, runs the algorithm, and outputs:

$$f(c) = \begin{cases} 1 \text{ if coin flip sequence } c \text{ leads to success} \\ 0 \text{ if coin flip sequence } c \text{ leads to failure} \end{cases}$$

- Can create a quantum alg that searches for the coin flip sequence that causes alg to succeed.
- If p fraction of coinflips marked:
  - Classical Probabilistic Query Complexity to f:
  - = Quantum Query Complexity to f:

ex: Best classical 3-SAT alg:

↳ Grover → Quant. Alg with runtime:

Can do for any alg.

Note

Not as clear for data...