Previously

- Berstein–Vizirani
- Simon's problem
- Period finding
- Shor's order finding
- Question: What do these tasks have in common?
  - Common generalization
  - Solve efficiently on a quantum computer?
  - Useful for anything interesting?

Groups

- Definition: A set G and a binary operation $\circ\colon\ G \times G \to G$ such that
  - Associativity: $x \circ (y \circ z) = (x \circ y) \circ z$
  - Identity element: there exists $e \in G$ such that $e \circ x = x = x \circ e$ for every $x \in G$
  - Inverse: for every $x \in G$, there exists $y \in G$ such that $x \circ y = y \circ x = e$
  - (Can show that such y is unique, and use x^{-1} in place of y)
- Then G together with $\circ$ is called a *group*.
- If $x \circ y = y \circ x$ for every $x, y \in G$, then G is a *commutative group* (or an *abelian group*).

Examples of groups

- $G = \{0,1\}^n$
- $G = \mathbb{Z}/N\mathbb{Z}$
- $G = (\mathbb{Z}/N\mathbb{Z})^*$
- $G = \mathbb{Z}/N\mathbb{Z} \times \mathbb{Z}/N\mathbb{Z}$
- Fact: every finite commutative group is “isomorphic” to

$$\mathbb{Z}/N_1\mathbb{Z} \times \mathbb{Z}/N_2\mathbb{Z} \times \ldots \times \mathbb{Z}/N_k\mathbb{Z}.$$

- Infinite commutative groups: $\mathbb{Z}, \mathbb{R}$

Non-commutative examples

- Symmetric group S_n
  - G is the set of all permutations of $\{1,\ldots,n\}$, that is, all bijections $\pi\colon \{1,\ldots,n\} \to \{1,\ldots,n\}$
  - $\circ$ is composition, that is $\pi_1 \circ \pi_2$ means "do π_2 then do π_1"
  - $e =$ ...?
  - π^{-1} means ...?
- Not commutative. Why?

Non-commutative examples

- Dihedral group D_n
  - G permutations $\pi\colon \{1,\ldots,n\} \to \{1,\ldots,n\}$ that are automorphisms of the cycle graph of length n
- Fact: G contains n reflections, $n-1$ rotations, and one identity.
- Example: $n = 4$

Subgroups

- Definition: Given a group G with $\circ$, a subset H of G is a subgroup of G when H together with $\circ$ also form a group.
- Take an element $h \in G$, subgroup generated by H is …?
- Example: $G = \mathbb{Z}/16\mathbb{Z}$ and $h = 4$
- Can also generate subgroup by two elements $h_1, h_2 \in G$.
- Example: Dihedral group D_n is a subgroup of S_n generated by …?

Cosets

- Definition: Suppose H is a subgroup of G, the left-coset of H with representative $x \in G$, denoted by xH, is $\{xh : h \in H\}$.
- Examples: $G = \mathbb{Z}/16\mathbb{Z}$ and H is generated by 4. What are the cosets?
- Facts:
    - $|xH| = |H|$
    - Any two cosets xH and yH are either identical or disjoint
    - The cosets of H partition G

Hidden subgroup problem

- Definition: A labeling F of G is *H-periodic* if F has the same label on all elements in xH for each coset xH, and F gives different cosets different labels.

- Hidden subgroup problem for G

  - Givne a quantum circuit implementing $F: G \to \{0,1\}^m$

  - Promised F is H-periodic, where H is a secret subgroup of G

  - Find H or a set of generators of H

- Example: Berstein–Vazirani, Simon's problem, Period-finding over $\mathbb{Z}/N\mathbb{Z}$

Solve HSP quantumly

- Step 1: Prepare uniform superposition

$$\frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle$$

- Step 2: Load “data” F:

$$\frac{1}{\sqrt{|G|}} \sum_{x \in G} |x\rangle |F(x)\rangle$$

- Step 3: Measure answer qubits, and get some label $C*$.
    - State collapses to …?

- Get a random coset state $|gH\rangle$ = …?
- Recall: a probability distribution over quantum states is called a mixed state
- ρ_H := uniform distribution over all coset states $|gH\rangle$
- Question: can we learn H from ρ_H?
- Idea:
  - Apply the appropriate Fourier transform for G and measure
  - Obtain a "clue" about H
  - Deduce H (hopefully) from the clues

- Fact 1: When G is finite commutative, that is $G = \mathbb{Z}/N_1\mathbb{Z} \times \ldots \times \mathbb{Z}/N_k\mathbb{Z}$, the appropriate Fourier transform is $\mathrm{DFT}_{N_1} \otimes \ldots \otimes \mathrm{DFT}_{N_k}$, which can be implemented efficiently by a quantum circuit.

- Fact 2: When G is not commutative, the appropriate Fourier transform can be implemented efficiently in most cases, but don't know how to deduce H from clues efficiently.