# Quantum Computing (Fall 2013) Instructor: Shengyu Zhang.

# Lecture 4 Hidden Subgroup Problem 2: Fourier sampling and query-efficient algorithms.

## HSP: What and why.

Suppose that is a group and is a subgroup. Recall that cosets of partition the group . A function *hides* a subgroup if for and in the same coset and for and in different cosets.

**Definition**. Given a black box function (for some finite set ) that hides a subgroup , the Hidden Subgroup Problem (HSP) asks to find .

Two important special cases: symmetric group and dihedral group.

Symmetric group : the set of permutations on elements. An efficient algorithm for HSP for symmetric group can be used to solve Graph Isomorphism in time.

Graph Isomorphism (GI): Given two -node graphs and , decide whether they are isomorphic, i.e. whether there exists a permutation s.t. .

Here for a graph where . Thus the two graphs are isomorphic if the following holds: iff .

Graph Isomorphism is known to be equivalent to the following variants: Graph Isomorphism Finding (given two isomorphic graphs, find an isomorphism), Graph Isomorphism Counting (given two isomorphic graphs, count the number of isomorphisms), Graph Automorphism Finding (given a graph , find its automorphism group ). The decision version of Graph Automorphism (just to decide whether ) is a clearly no harder, and possibly easier task.

Graph Isomorphism is a fundamental problem that appears in many disciplines, and it’s one of the few problems whose complexity isn’t pinned down: It’s not known to be in P, but it’s also not known to be NP-complete either. Actually, most people believe that it is not NP-complete. It may be well in P and we just haven’t found an algorithm yet.

Given the equivalence between GI and Graph Automorphism Finding (GA), we can see why GI reduces to HSP for symmetric group. First reduce GI to GA. Now for GA, given a graph , define on by . We claim that hides . Actually,

The second major application of HSP for non-Abelian groups is the *(Approximate) Shortest Vector* problem, which reduces to HSP for dihedral group. We’ll define (Approximate) Shortest Vector problem and dihedral group next.

Consider and a set of basis, namely linearly independent vectors in it. An -*dimensional lattice* is the set of all **integer** linear combinations of basis vectors. In the Shortest Vector problem, we want to find a shortest (but nonzero) vector in the lattice. That is, we want to use integer linear combination of basis vectors to get a vector that is very close to the origin. This problem itself is NP-hard, and we consider a relaxed version, that the input is promised to have only one (up to its sign) vector that achieves the minimum length; all the other vectors are at least times longer. Of course, the larger is, the stronger the promise and thus the easier the problem. It is known that if then the problem is still NP-hard, and if , then the problem becomes in P. The question is what the complexity is for . It is conjectured to be hard, and actually cryptosystems are built based on this computational assumption. What can a quantum computer do? If HSP for dihedral group is easy, then there is an efficient quantum algorithm to solve the (Approximate) Shortest Vector problem. The reduction is nontrivial, and we won’t do it here. (It’s actually one of the reading projects.) We’ll just introduce what a dihedral group is.

Consider a regular -gon on a 2D plane. A symmetry is a rotation or a reflection which keeps the -gon unchanged. A regular -gon has symmetries: rotational ones and reflection ones, the collection of which form the dihedral group . So

( is generated by elements and satisfying the relations .)

## Detour: density matrices.

Mixed states: Pure states with probabilities.

* , rank-1 positive semi-definite (psd) matrix
* , trace-1 psd matrices.
**Fact**: Any trace-1 psd matrix also corresponds to an ensemble of quantum pure states.
*Proof*. Do the spectral decomposition and use the fact that the trace is 1 to conclude that the sum of eigenvalues is 1.
* Unitary transform: .
* Measurement: with . Then , and the post-measurement state is .
Why use density matrices: Because only those matter---The exact ensemble of pure states doesn’t since we can’t distinguish different ensembles with the same density matrix.
* Composition: .

## Standard approach, weak and strong Fourier sampling

**Standard approach**:

 for a random

Writing this in the density matrix form, we have the following

where is a set of representatives.

**Weak Fourier sampling**:

 observe w.p.

**Question**: Does samples of contain enough info to determine ?

For Abelian groups: Yes, as previously discussed.

Another class of subgroups that weak Fourier sampling suffices: normal subgroups.

**Definition**. A subgroup is *normal* if , for all .

So if is normal subgroup, then for any , for some . And if runs over , then so does .

**Theorem**. If is a normal subgroup of , then the weak Fourier sampling gives with probability if , and 0 otherwise.

*Proof*. If , then one observes w.p. .

If is not contained in , then s.t. . Now note that

thus by Schur’s lemma, for some . But

Thus So one observes w.p. 0.

The algorithm for HSP for normal subgroup is very similar to that for Abelian HSP.

* Use weak Fourier sampling to get for .
* Output

**Theorem**. with high probability.

*Proof*. If , we claim that Indeed,

Here we used a fact that for any normal subgroup of , .

**Strong Fourier sampling**:

Weak Fourier sampling fails to provide sufficient information for HSP for and . So people resort to strong Fourier sampling, which uses the remaining state .

**Question**: What basis to use to measure this state?

It turns out that even random basis can already solve some cases, such as HSP for Heisenberg Group

However, for symmetric group, even strong Fourier sampling fails. Multi-register measurement is needed.

## Query efficient algorithm

When multi-register measurement is used, we can solve HSP using only queries to , though the computational time is still exponential.

Recall that the standard approach gives . Our task is just to identify from a collection . In general, mixed state identification is known to have the following bound.

**Theorem**. There exists a quantum measurement that identifies with probability at least .

Here is a measure of how close two mixed states and are. The value is always between 0 and 1, and the closer to 1, then closer the two states. So to identify (namely to identify ), it is enough if ’s for different are far from each other. It turns out that this is more or less true.

**Theorem**. .

We will not prove this result, but only explain how to use it. It is not hard to verify that

By the above results, we know that if we want the error probability to be , then we need to use copies of , satisfying

Solving this gives . How large is ? It’s just the number of subgroups.

**Fact**. Any group has at most subgroups.

*Proof*. Each subgroup has a generating set of size at most , because adding one more generator at least double the size of the subgroup. Therefore, the number of subgroups is at most .

## Reference

[CvD10] Andrew M. Childs and Wim van Dam, Quantum algorithms for algebraic problems, *Reviews of Modern Physics*, Volume 82, January–March 2010.

[Lom04] Chris Lomont, The hidden subgroup problem - review and open problems, arXiv:quant-ph/0411037v1