1 Results

1.1 No-go theorem for speedup in PIP

Problem 1 (Point-In-Polygon)**.** *Given a point $p \in \mathbb{R}^2$ and a polygon with n points named $q_1, \ldots, q_n \in \mathbb{R}^2$, we say* Yes *if p lies inside the polygon and* No *otherwise.*

Theorem 1. *Quantum query complexity of problem 1 is $\Omega(n)$.*

Fact 1. *There is a classical algorithm that can solve problem 1 using $\Theta(n)$ queries in $\Theta(n)$ time.*

1.2 Superpolynomial speedup in PIPP

Problem 2 (Point-In-Polygon-Promise)**.** *Same as problem 1, but it's promised that the apparent angle θ_i of each edge $e_i := (q_i, q_{i+1})$ from the viewpoint of p is less than $\frac{2\pi\kappa}{n}$, where κ is an $O(1)$ constant.*

Theorem 2. *Classical query complexity for any bounded-error algorithm that solves problem 2 is $\omega(1)$.*

Theorem 3. *There is a bounded-error quantum algorithm that can solve problem 2 using $\Theta(1)$ queries in $\Theta(\log(n))$ time.*

$$\Pr(0) = \left| \langle 0| H^{\otimes \log(N-1)} |\psi\rangle \right|^2 = \begin{cases} \frac{4\kappa^2\pi^2}{(N-1)^2} & \textit{in} \\ 0 & \textit{out} \end{cases} \tag{1}$$

2 Proofs

Problem 3 (Parity)**.** *Given a bitstring consisting of n bits, we say* Yes *if the hamming weight of the bitstring is even, otherwise* No*.*

Lemma 1. *Problem 3 can be reduced to problem 1 with no query overhead.*

Proof. Assume a regular polygon with n points $q_1, \ldots, q_n$, naming its center point p. We can map each point to each bit. If the bit is equal to 1, we reflect q_i and then scale by $\alpha \neq 1$ with respect to p.

Now, solving problem 1 for the polygon will solve problem 3. Note that for each point query, we need to do exactly 1 bit query. □

Proof of theorem 1. First, We prove a lower bound for problem 3. Using quantum adversary bound [**?**], we can create a hypercube associated with the bitstring that can be seen as a bipartite that one part is the set of bitstrings with output Yes in problem 3 and the other part with No. The degree of nodes is equal to n on each part, and defining subgraphs G_i for edges associated with i-th bitflip, they'll have node with degree equal to 1.

Then the lower bound will be

$$\Omega\left(\sqrt{\frac{mm'}{ll'}}\right) = \Omega(N)$$

We can use lemma 1 to generalize the lower bound to problem 1. $\square$

Definition 1. *Assume a regular polygon with n points and a center called p. We define cresent transformation as if we transform $k \leq n/2$ sequential edges to make a cresent shape and make p out of the polygon. It's easy to show that $\kappa = \max \theta_i$ (as it's defined in 2) in this transformation will change from 1 to $\frac{n-k}{k}$*

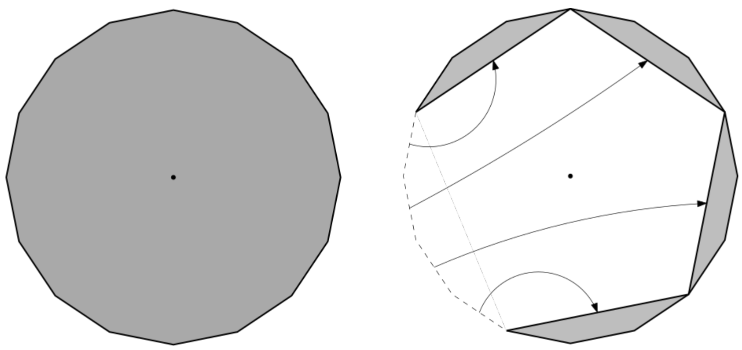

Figure 1: Illustration of lemma 1

Proof of theorem 2. Assume a regular polygon with n points and a center called p (as in definition 1). If we use t queries, where $t \in \Theta(1)$, there are $\lceil \frac{n-t}{t} \rceil + 1 \in \Theta(n)$ sequential edges are left in between. These edges are totally unkown (note that two ends are fixed). Then we can apply cresent transformation on the sequential edges that results in a new polygon, that the following statements are all true for them.

- The old one and the new one cannot be distinguished with those queries.
- The promise $k \in O(1)$ is valid for both polygons.
- The result of problem 2 differs for these two polygons.

This means that t queries aren't sufficient to solve this problem and we need $t \in \omega(1)$ queries. $\square$

Proof of theorem **??**. $\square$