# Quantum algorithms and lower bounds for 3-Sum problem

Oksana Scegulnaja[1] * Andrej Dubrovsky[1] †

[1] *Institute of Mathematics and Computer Science, University of Latvia, Raina bulv. 29, Riga, Latvia.*

**Abstract.** In this work we observe a fundamental problem in Computational geometry that is called 3-Sum. We present quantum algorithm, that solves this problem in $O(NLogN)$ time, and lower bound $\Omega(\sqrt{N})$, although very low and unreachable. We also discuss generalization of 3-Sum called r-Sum and present algorithm, that works within $O(N^{\lceil r/3 \rceil} logN)$ and lower bound $\Omega(\sqrt{N})$.

## 1 Introduction

There is a big cluster of problems in computational geometry, that are called 3-Sum Hard. Gajentaan and Overmas [1] described them as problems, that can be reduced to the 3-Sum problem. The original notation of the 3-Sum problem is the following. Given the set S of N numbers, detect whether there are three numbers $a \in S, b \in S, c \in S$, such that a + b + c = 0. There are some other notations, that are equivalent to the first one, used in the literature.

In classical computation the best currently known algorithm for any 3-Sum Hard problem takes $O(N^2)$ time, while the best lower bound for the time complexity is $\Omega(NlogN)$, which is very low and unreachable. It is believed that 3-Sum lower bound is the same as upper bound, $\Omega(N^2)$, so 3-Sum hard problems in classical computation sometimes call $N^2 - hard$. For some of 3-Sum hard problems $\Omega(N^2)$ lower bound has been proved.

## 2 Preliminaries

We will not describe here general model of quantum computation. To get basic information on the specific topic please refer to J. Gruska [2].

We consider the following problem, called 3-Sum problem:

**Definition 1** *Given the 3 sets A, B and C each of N numbers, detect whether there are three numbers $a \in A, b \in B, c \in C$, such that a + b = c.*

We are interested in finding quantum time complexity of 3-Sum algorithm. In our algorithms we make use of quantum amplitude amplification method [3], which generalizes Grover quantum search.

We also use quantum adversary method [4], [5] to compute lower bound for our task.

## 3 3-sum problem

**Theorem 2** *3-Sum problem has quantum lower bound $\Omega(\sqrt{N})$*

*Proof.* The proof is based on A. Ambainis adversary method of proving quantum lower bounds [4], [5]. We use his main theorem.

We observe a function $f(x_1, x_2, ..., x_{3N})$, which returns 1 if 3-sum problem has an answer, and two sets of inputs $X$ and $Y$ such that $f(x) \neq f(x)$ if $x \in X$ and $y \in Y$. We need to find all the variants how the input $X$ can be modified, to change the value of the function. For our case, we'll take X to contain only one element: A consisting of all zeros; B - all 1s, C - all 2s. It is an input, on which our function returns 0. Let Y contain all inputs made of X, with one element in any of sets A, B and C changed so, that the function returns 1 (e.g., any element of A changed to 1; any element of B - to 2; any element of C - to 1). Let $R \subset X \times Y$ consist of such $X \times Y$ pairs where $y$ differs from $x$ in exactly one position. According to the theorem, $m$ = 3N, because for every $x \in X$ there are exactly 3N different $y \in Y$, which differs from $x$ in exactly one position. $m' = 1$, because for every $y \in Y$ there is only one $x \in X$, which differs from $y$ in exactly one position. $l = l' = 1$ that follows from our definition of $R$.

Using this formula we get a lower bound $\Omega(\sqrt{3N})$.

□

**Theorem 3** *There exists a quantum algorithm that solves 3-Sum problem in O(NlogN).*

*Proof.* The algorithm works as follows:

1. Sort set C classically, that takes O(NlogN) time.
2. Construct an algorithm that can solve the problem with small probability. The algorithm takes an input of two random elements, one from set $A$ and the other from set $B$ and outputs whether these two elements are summing up to some element from $C$:
   (a) Compute a+b, $a \in A, b \in B$.
   (b) Check whether a+b can be found in C. Needs $O(logN)$ time, because C is sorted.
3. Construct quantum superposition over all the $|a\rangle|b\rangle$ and use amplitude amplification on that superposition with the algorithm just described as a kernel.

Amplitude amplification method uses Grover algorithm idea to speed up computation. The maximum speedup it allows to get is quadratic. In our case, classically we must repeat algorithm kernel steps $O(N^2)$ times. Amplitude amplification method allows us to get the same result, using only O(N) steps.

*oksana.s@liis.lv

†a.dubrovskis@alise.lv

So the total time the algorithm uses is O(NlogN). $\square$

## 4 3-sum generalization

The most natural generalization of 3-Sum problem is its r-Sum, as defined in [6]:

**Definition 4** *Given r sets $S_1, ..., S_r$ of N numbers, detect whether there are r numbers one from each set that sum to zero.*

Similarly we can define the class of r-Sum Hard problems.

In deterministic case these problems have a lower bound of $\Omega(Nlog(N))$ and best known deterministic algorithm can solve the r-sum problem in $O(N^{r+1/2})$ when r is odd and $O(N^{r/2}log(r))$ when r is even.

**Theorem 5** *Quantum algorithm can solve r-Sum problem in $O(N^{\lceil r/3 \rceil}logN)$ time.*

*Proof.* We will further divide sets $S_1, ..., S_r$ in two more groups of sets: First group will contain x sets of data - let's call them $C_1, ..., C_x$ and the second will contain remaining (r-x) sets of data - let's call them $Q_1, ..., Q_{r-x}$.

The algorithm itself consists of two parts - classical and the quantum one. First, we execute the classical part of the algorithm and then execute the quantum part of it.

Classical part of algorithm works as follows: we take sets $C_1, ..., C_x$ and perform following operations on them:

1. Make all the possible groups of elements picking one from each of sets $C_1, ..., C_x$ and sum them up. (Call this new set CSum)

2. Sort the summary set CSum

After classical part of algorithm finishes its work we will get the sorted set of all the possible element combinations in sets $C_1, ..., C_x$. This will take $O(N^x logN)$ time.

Then starts the quantum part of the algorithm, that uses sets $Q_1, ..., Q_{r-x}$ as well as the set CSum. The quantum part of r-sum algorithm is in fact generalized version of quantum 3-sum algorithm:

1. Construct the probabilistic algorithm that can find the solution of the problem with a small probability. The algorithm will take two steps: This algorithm will randomly take one element from each of sets $Q_1, ..., Q_{r-x}$ and sum those elements. Then it will take the sum obtained and search for it in CSum sorted set.

   The first algorithm step can be accomplished in constant time that depends only on r. The second algorithm part takes $O(log(N))$ time to search element in sorted database. So the total running time of the algorithm is $O(log(N))$. However this algorithm will find the solution with probability only $\frac{1}{N^{r-x}}$.

2. To boost the probability of success we use amplitude amplification technique. We prepare the starting quantum superposition $\sum |Q_1\rangle|Q_2\rangle...|Q_{r-x}\rangle$ and call the amplitude amplification with algorithm from the step 1 embedded in it. Then, after $O(N^{(r-x)/2})$ steps the algorithm will give an answer with high probability of success. So the running time of quantum part of algorithm is $O(N^{(r-x)/2}logN)$.

The total running time of this algorithm is: $O(Max(N^x log(N), N^{(r-x)/2}log(N)))$, so we should minimize this function. This is the case when x = (r-x)/2. So we get the value of x = 1/3r and total algorithm running time $O(N^{\lceil r/3 \rceil}logN)$. $\square$

**Theorem 6** *r-Sum problem has quantum lower bound $\Omega(\sqrt{N})$*

*Proof.* The proof is the same as for 3-Sum problem. $\square$

## 5 Conclusion

In this paper we have observed 3-Sum problem. We have created quantum algorithms from 3-Sum problem and its generalization called r-Sum, that run faster that their classical counterparts. As to lower bounds, we have found the same gap between the best lower bound we can obtain and intuitively optimal, as in classical case both for 3-Sum and r-Sum.

## References


[1] A. Gajentaan, M. H. Overmars. On a class of $o(n^2)$ problems in computational geometry. *Computational Geometry: Theory and Applications, 5(3)*, pages 165–185, 1995.

[2] J. Gruska. *Quantum Computing*. McGraw-Hill, 1999.

[3] G. Brassard, P. Hoyer, A. Tapp. Quantum algorithm for the collision problem. *ACM SIGACT News (Criptology column)*, 28:14-19, 1997. quant-ph/9705002.

[4] A. Ambainis. Quantum lower bounds by quantum arguments. Manuscript, 1999.

[5] A. Ambainis. Quantum query model: algorithms and lower bounds. *Proceedings of the Second International Workshop on Quantum Computing and Learning*, 1999.

[6] J. Erickson. New lower bound for convex hull problems in odd dimensions. *Proceedings of the twelfth annual symposium on Computational geometry*, pages 1-9. ACM Press, 1996.