CMSC 858M: Algorithmic Lower Bounds: Fun with Hardness Proofs
Fall 2020

Instructor: Mohammad T. Hajiaghayi
Scribe: Jacob Prinz

TBD

1 Overview

In complexity theory, one often deals with large classes of time complexity, like P, the set of all polynomial time functions. However, in some applications, one may want to deal with more specific time complexity classes. In this section, we present the 3SUM problem. This problem is known to be able to be solved in quadratic time. It has been shown to be equivalent to a number of other problems. None of these problems has a known solution in sub-quadratic time (which we will define in this section). Therefore, we conjecture that 3SUM, and therefore many other problems, can not be solved in sub-quadratic time. One may then use this assumption to demonstrate the hardness of other problems by reducing them to 3SUM.

2 3SUM

Definition 1 (3SUM) *Given a set of n integers, do any three of the integers sum to zero?*

Clearly, this problem can be solved in $O(n^3)$ time by simply trying every subset of three integers.

Theorem 1 (Theorem) *3SUM can be solved in $O(n^2 \log n)$ time*

Compute all sums of subsets of size two. Sort the set of integers in $O(n \log n)$ time. Then for each pair sum, do binary search on the sorted list to find if there is a third integer adding to zero.

Theorem 2 (Theorem) *3SUM can be solved in non-deterministic $O(n^2)$ time.*

Start by finding all of the pair sums as before. Then, for each pair sum, negate the sum to get what the third number would have to be. Then, put all of the pair sums in a hash table where the keys are based on that hypothetical third number. Then, for each number in the set, check if it is in the hash table. This on average takes $O(n^2)$ time, but hypothetically many numbers could end up in one bucket, so the time could be worse.

Theorem 3 (Theorem) *3Sum can be solved in* $O(n^2)$ *time deterministically,*

Sort the numbers in $O(n \log n)$ time. Then, for each number in the list, make a new sorted copy without that number. Make a pointer to the top and bottom of the list. While the current sum is too largs, move the top pointer down. While the current sum is too small, move the bottom pointer up. When the pointers collide, that means there is no triple summing to zero for that given first number. To see that this finds the triple if it exists, suppose that the number picked is one in the triple. Suppose that the bottom pointer has been moved to another one of the numbers in the triple. Then, if the top pointer is too high, it will be moved down until the triple is found. The top pointer can't be too low, because once it is at the correct number, the sum will always be too low until the bottom pointer moves to the correct location.

3 Subquadratic Time

A natural question is if there is an algorithm that does better than quadratic time. The answer depends on your definition of "better". For example, Gronlund and Pettie found an algorithm that runs in $O\left(\frac{n^2}{(\frac{\log n}{\log \log n})^{\frac{2}{3}}}\right)$ [?].

Definition 2 *An algorithm runs in* subquadratic *time if there exists a number* ϵ *such that it runs in* $O(n^{2-\epsilon})$ *time.*

To date, there is no known algorithm for 3SUM which runs in subquadratic time. Therefore, computer scientists conjecture that there does not exist an algorithm for 3SUM in subquadratic time. Other results can then be based on this conjecture.

Definition 3 *Let* A *and* B *be decision problems. Then* $A \leq_{sq} B$ *means that*

$$B \text{ subquadratic} \rightarrow A \text{ subquadratic}$$

Furthermore, define $A \equiv_{sq} B$ when $A <_{sq} B$ and $B <_{sq} A$.

Definition 4 *A problem A is* 3SUM hard *when* $A \equiv_{sq} 3SUM$.

4 Variants of 3SUM

In this section, we list several problems which are very similar to 3SUM, and are known to be 3SUM hard.

The first is due to Patrascu [?].

Definition 5 (Convolution 3SUM) *Given a set of n integers* $a_1, \dots a_n$*, are there two number* $i < j \leq n$ *such that* $a_i + a_j = a_{i+j}$*?*

Clearly this can be solved in quadratic time, but Patrascu's contribution was to show that this is 3SUM hard.

The following definition has no particular name, but is interestsing in illustrating the hardness of 3SUM

Definition 6 (variant of 3SUM) *Given a set of* n *integers between* $-n^3$ *and* n^3*, do any three sum to zero?*

This variant is also 3SUM hard.

The following is due to Gajentann and Overmars [?] and is also 3SUM hard.

Definition 7 (3SUM') *Given three sets of n integers* A*,* B*, and* C*, Is there an* $a \in A$*,* $b \in B$*, and* $c \in C$ *such that* $a + b = c$*?*

5 Computational Geometry Problems that are 3SUM hard

The following is due to Gajentaan and Overmars [?]

Definition 8 (GEOMBASE) *Given* n *points in* $\mathbb{Z}^2$ *with* y*-coordinate in* $\{0, 1, 2\}$*, does there exist a non-horizontal line hitting 3 points of this set?*

Theorem 4 *GEOMBASE is 3SUM hard.*

We will proceed by showing that $\mathsf{3SUM}' \leq_{sq} \mathsf{GEOMBASE}$.

Given an input A, B, C to 3SUM', construct the following GEOMBASE problem:

$$\{(a, 0) : a \in A\} \cup \{(b, 2) : b \in B\} \cup \{(\frac{c}{2}, 1) : c \in C\}$$

Clearly, a middle point lies on a line with a top and bottom point if its x-coordinate is the average. $\frac{c}{2}$ is the average of a and b iff $a + b = c$.

Theorem 5 *3SUM'* $\leq_{sq}$ *GEOMBASE*

The following variant will also be useful:

Theorem 6 (GEOMBASE') *Given n points in $\mathbb{Z}^2$ with y-coordinate in $\{0, 1, 2\}$, and ϵ, take the points as ϵ-long holes in the lines $y = 0$, $y = 1$, and $y = 2$. Then, is there a non-horizontal line passing through the holes?*

The following is due to Gajentaan and Overmars [?]

Definition 9 (COLLINEAR) *Given n points in $\mathbb{Z} \times \mathbb{Z}$, are any three of the points collinear?*

Theorem 7 *COLLINEAR is 3SUM hard.*

This time, we will proceed by showing that $3\mathsf{SUM} \le \mathsf{sq}\mathsf{COLLINEAR}$.

Given an instance of $3\mathsf{SUM}$, which is a set of integers A, take the set $\{(x, x^3) : x \in A\}$. Algebra shows that three of these points are collinear if and only if their sum is zero. Refer to a homework problem for more details.

Definition 10 (CONCURRENT) *Given n lines, do any three of them intersect in a point?*

In geometry, there is a duality between points and lines. If all lines $ax + by + 1 = 0$ are swapped for the point (a, b) and vice versa, then every point intersecting with a line turns into a line intersecting the point. Therefore,

$$\text{COLLINEAR} \le_{sq} \text{CONCURRENT}$$

Definition 11 (SEP) *Given n line segments in a plane, is there a line which does not intersect any segment, and there is at least one segment on each side of the line?*

Theorem 8 *GEOMBASE'* $\le_{sq}$ SEP

Given a instance of $\mathsf{GEOMBASE}'$, which is a set of points A and a number ϵ, we are left with three rows of line segments. The existence of a separator is then equivalent to the existence of a line passing through three holes.

6 Strips

Definition 12 *A* strip *is the region between two parallel lines.*

The following is due to Gajentaan and Overmars [?]

Definition 13 (STRIPS) *Given n strips and an axis-aligned rectangle, can a union of the strips cover the rectangle? The strips have a fixed width, but can be rotated and scaled.*

Theorem 9 *STRIPS is 3SUM hard.*

We will show that GEOMBASE' $\leq_{sq}$ STRIPS.

Given an instance of GEOMBASE', we have a set of vertical line segments. Consider the point-line duality where the point (a, b) is mapped to the line $x = ay + b$ and vice verse. Then, parallel lines are mapped points with the same a-coordinate. Therefore, the horizontal line segments from GEOMBASE' correspond to strips. Each point in the rectangle corresponds to a non-horizontal line. Therefore, there is a point in the rectangle uncovered by strips if and only if there is a line passing through the GEOMBASE' instance. Therefore, we can use STRIPS to solve GEOMBASE'.

Definition 14 (Triangle Cover Triangle (TCT)) *Given a set of n triangles and a target triangle, can the set of triangles cover the target?*

Theorem 10 *Triangle Cover Triangle is 3SUM-hard.*

We will proceed by showing that STRIPS $\leq_{sq}$ TCT.

Given an instance of STRIPS, we can triangulate the box and the portions of the strips which are within some bound of the box. Then we have reduced STRIPS to a number of TCT problems. Each thing only takes two triangles, so this retains the complexity.

Definition 15 (Hole in Union (HIU)) *Given a set of n triangles, is there a closed region enclosed by the triangles?*

Theorem 11 *TCT $\leq_{sq}$ HIU*

Take an instance of TCT. Add more triangles that cover area outside target traingle, as well as edges of target. Then, there is a hole if and only if the original triangles did not cover the target.

Definition 16 (Triangle Measure Problem (TM)) *Given n triangles, what is the area of their union?*

Theorem 12 *TM is 3SUM-hard*

We proceed by showing that TCT $\leq_{sq}$ TM. Suppose that we have a TCT instance. Then, add triangles to the outside so that we know that all space in a large rectangular area around all of the traingles, but not including the target interior, is filled. Then, the total area is equal to the area of that rectangle if and only if the target is covered.

Definition 17 (Point Covering (PC)) *Given a number k and n half planes, is there any point covered by k of the half planes?*

Note that if $k \leq \frac{n}{2}$, then the answer is always yes.

Theorem 13 *PC is 3SUM-hard*

We proceed by reduction of PC to STRIPS. For each strip, make two half planes which cover the complement of the strip. Then, each point either lies in a strip, or lies in n half-planes. Next, make four half planes which cover the inside of the rectangle from the STRIPS problem. Then, there is a point in the rectangle not covered by a strip iff there is a point the is in $n + 4$ half planes.

We proceed by showing that STRIPS $\leq_{sq}$ PC.

Given an instance of STRIPS, for each strip, take the two half-planes which cover the outside of the strip. Then, any point which does not lie under any strips lies in n half-planes. Add four more half-planes to outline the rectangle. Then, if there is a point covered by exactly $n+4$ planes out of the $2n+4$ total, then it is in the rectangle.

7 Visibility Problems

Definition 18 (Visibility Between Segments (VBS)) *Given n horizontal line segments, as well as two more horizontal segments s_1 and s_2, is there a segment connecting s_1 and s_2 which does not intersect any of the n horizontal segments?*

Theorem 14 *VBS is 3SUM-hard.*

We proceed by reduction of VBS to GEOMBASE'. GEOMBASE' involves horizontal segments and a question of if there is a line not intersecting them. We can instead place s_1 and s_2 above and below all of the segments, and ask if there is a segment between s_1 and s_2 not intersecting any other segments.

Definition 19 (Visible Triangle Problem (VT)) *Given n horizontal triangles in $\mathbb{Z}^3$, and a particular horizontal triangle T, and a point in $\mathbb{Z}^3$, is there a line segment from the point to T that does not pass through any of the n triangles? The triangles are solid.*

Theorem 15 *VT is 3SUM-hard*

We proceed by reducing TCT to VT. Make the point very far above T, or even at infinity, and the other triangles close to T. Then, the segment exists iff the triangles do not cover T when projected onto its plane.

Definition 20 (Planar Motion Planning Problem (PMP)) *Given n line segments, each either horizontal or vertical, and two points $\mathsf{thesource}$ and $\mathsf{thegoal}$, can a line segment be translated and rotated to move from $\mathsf{thesource}$ to $\mathsf{thegoal}$ without hitting any of the obstacles?*

Theorem 16 *PMP is 3SUM-hard*

We proceed by reduction of GEOMBASE' to PMP. Simply make the robot a long segment, and it can pass through the GEOMBASE' problem iff there is a straight line through it.

8 Fixed Angle Chains

The results in this section are due to Soss et al. [**?**].

Definition 21 *A* fixed angle chain *is a chain of line segments such that the angle between adjacent line segments is fixed, but can be flipped one direction or the other.*

Definition 22 (Fixed Angle Chain Problem (FAC)) *Given a fixed angle chain, can any angle be flipped to cause a collision?*

Theorem 17 *FAC is 3SUM-hard*

We proceed by reducing 3SUM' to FAC. Suppose we have our instance of 3SUM', which are sets A, B, and C. The question is if any triple from these three sets adds in a certian way. Note that any linear scaling of A, B, and C results in a similarly hard problem.

One can construct a fixed angle chain by first starting with two horizontal segments one $n/3$ units apart vertically. Place some spikes in these corresponding to the values in A and C. Then, connect them with a staircase, where the horizontal positions of the stairs corresponds with B. Then, the only angles which could be flipped to causes a collision are the verticals on the stairs. Then, spikes intersect if and only if something in A, B, and C adds up in a certian way.

9 Other Lower Bounds

There are some other lower bounds that have been derived from 3SUM hardness [**?**].

Definition 23 (Problem statement:) *Given a weighted graph of size n, is there a traingle with weight n?*

Theorem 18 *There is an $O(n^{3/2})$ algorithm*

Theorem 19 *If there is an $O(n^{3/2} - \epsilon)$ algorith, then there is a subquadratic algorithm for 3SUM.*

Definition 24 (Problem statement:) *Given a graph, are there n triangles?*

Theorem 20 *There is an $n^{3/2}$ algorithm, but if there is an $O(n^{4/3} - \epsilon)$ algorith, then there is a subquadratic algorithm for 3SUM.*

10 dSUM

There is a natural generalization of 3SUM:

Definition 25 (dSUM) *Given a set of n integers, do any d of them add to 0?*

The following is due to Patrascu and Williams [**?**]:

Theorem 21 *Let $d < n^{.99}$. If dSUM with numbers of $O(d \log n)$ bits can be solved in $n^{o(d)}$ time, then 3SAT can be solved in $2^{o(n)}$ time.*

This strongly suggests the hardness of $dSUM$.

11 Extra Related Problems

Barequet and Har-Peled [**?**] proved the 3-SUM hardness of some problems in computational geometry. They proved that each of the following problems is 3SUM-hard:

Definition 26 *Let P and Q be any two simple polygons. Determine if P can be translated to fit inside Q.*

Definition 27 *Let P and Q be convex polygons. Can P be translated and rotated to fit inside Q*

Definition 28 *Let P and Q be convex polygons. Can P be rotated around a given point to fit inside Q?*

Definition 29 (Segments Containing Points (SCP)) *Given a $P \subseteq \mathbb{R}$ and a set Q of intervals of real numbers, is there a $u \in \mathbb{R}$ so that $P + u \subseteq Q$?*

Also, Chan [**?**] got an algorithm for 3SUM in

$$O(\frac{n^2}{\log^2(n)}(\log \log n)^{O(1)})$$

Time

References

[1] Chan, T. M. (2020). More Logarithmic-factor Speedups for 3SUM, (median,+)-convolution, and Some Geometric 3SUM-hard Problems. ACM Transactions on Algorithms, 16(1), 1–23. https://doi.org/10.1145/3363541

[2] Gill Barequet, Seriel Har-Peled. Polygon Containment and Translational Min-Housdorff-Distance Between Segment Sets are 3SUM-Hard *International Journal of Computational Geometry & Applications, Vol. 11, No. 4 (2001)*

[3] Antonio Barrera, Algorithms for deciding the containment of polygons, *Information Processing Letters 59 (1996).*

[4] Erik Demaine, William Gasarch, Mohammad Hajiaghayi *Fun with Hardness: Algorithmmic Lower Bounds*. Not yet Published.

[5] Hajiaghayi, Mohammad T. *Algorithmic Game Theory: Lecture 1– Handwritten notes*. 1 September 2010. University of Maryland.

[6] Ilya Baran, Erik D. Demaine, and Mihai Patrascu. Subquadratic algorithms for 3sum. *Algorithmica*, 2008. https://doi.org/10.1007/s00453-007-9036-3.

[7] Allan Grønlund and Seth Pettie. Threesomes, degenerates, and love triangles. *Journal of the Association of Computing Machinery (JACM)*, 65(4):22:1–22:25, 2018. https://doi.org/10.1145/3185378.

[8] Nisan, Noam et al. *Algorithmic Game Theory*. New York: Cambridge University Press, 2007.

[9] Mihai Patrascu. Towards polynomial lower bounds for dynamic problems. *roceedings of the 42nd ACM Symposium on The- ory of Computing, STOC 2010, Cambridge, Massachusetts, USA* https://doi.org/10.1145/1806689.1806772.

[10] Anka Gajentaan and Mark H. Overmars. On a class of O(n 2) problems in computational geometry. Computational Geometry, 45(4):140–152, 2012. https://doi.org/10.1016/j.comgeo.2011.11.006.

[11] Ross, Don. *Game Theory (Stanford Encyclopedia of Philosophy)*. May 5, 2010. Stanford University. 6 September 2010 < http://plato.stanford.edu/entries/game-theory/ >.

[12] Feng, Xiaofan. *Cost-matrix drawing procedure for LaTeX*. The George Washington University. 6 September 2010.

[13] Mihai Patrascu and Ryan Williams. On the possibility of faster SAT algorithms. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January* 17-19, 2010, pages 1065–1075. SIAM, 2010. https://doi.org/10.1137/1.9781611973075.86.

[14] Michael A. Soss, Jeff Erickson, and Mark H. Overmars. Preprocessing chains for fast dihedral rotations is hard or even impossible. Computational Geometry, 26(3):235–246, 2003. https://doi.org/10.1016/S0925-7721(02)00156-6.

[15] Mihai Patrascu and Ryan Williams. On the possibility of faster SAT algorithms. In Moses Charikar, editor, *Proceedings of the Twenty-First Annual ACM-SIAM Symposium on Discrete Algorithms, SODA 2010, Austin, Texas, USA, January* 17-19, 2010, pages 1065–1075. SIAM, 2010.