3-Sum and quadratic hardness

→ So far we have seen APSP as a potential problem in $\theta(n^3)$.
→ Here we see another simple problem as a potential problem in $\theta(n^2)$.

3-sum: Given a set S of n integers, are there three elements of S with possible repetition that sum up to zero (we can additionally assume $S \subseteq \{-n^3, \dots, n^3\}$)

k-sum can be defined similarly.

→ A simple $\tilde{O}(n^2)$ algorithm as follows: take any two numbers $O(n^2)$ and search negate of their sum in the sorted list $O(\log n)$.

However we can obtain an algorithm of $O(n^2)$ as well.

Here we can define again

Subquadratic reduction from A to B:

Assumming an algorithm of truly subquadratic for B, we can obtain an algorithm of truly subquadratic for A.

3-sum conjecture: 3-sum is truly quadratic time, i.e., there is no algorithm of $n^{2-\varepsilon}$, for $\varepsilon > 0$.

→ Lots of people believe in 3-sum conjecture.

→ There are lots of hardness based on 3_sum conjecture for different fields: (e.g. see [Abboud & Williams' 14 FOCS] for Dynamic problem in which given a class of input objects, find efficient algorithms and data structures to answer a certain query about a set of input objects each time the input data is modified, i.e., objects are inserted or deleted. See several recent followups [AW14])

→ Also note that though truly-cubic and truly-quadratic algorithms are very important truly-linear are not at that level of importance for exact algorithms since we often need to read the input which is already $\Omega(n)$.

Now we give several 3-sum hard problems:

First 3SUM': Given three sets of integers A, B, C of total size n, are there $a \in A$, $b \in B$ and $c \in C$ with $a+b=c$?

First reduction from 3-sum to 3sum': simply set $A=S$, $B=S$ and $C=-S$.

From 3SUM′ to 3SUM: It is a bit more involved: Firs assume all elements in A, B, C are positive (otherwise add a large number K to A & B and $2K$ to elements of C). Let $m = 2\cdot\max(A, B, C)$ we construct S as follows: for each element $a \in A$ we put $a' = a + m$ in S. For each element $b \in B$, we put $b' = b$ in S and for $c \in C$ we put $-c-m$ in S. Clearly if $a+b=c$ then $a'+b'+c'=0$. What about the reverse: by construction $m < a' \leq 1.5m$, $0 < b' \leq 0.5m$ and $-1.5m \leq c' < -m$. Let $x, y, z \in S$ with $x+y+z=0$. At most one of the three elements can originally come from A because otherwise the sum would be larger than $2m - 1.5m = 0.5m$ which is larger than any element coming from B. Similarly only one element can come from C because otherwise the sum would be less than $-2m + 1.5m = -0.5$ (again the element coming from B cannot compensate it then. Also since all elements of A and B are positive at least one element should come from C. The only remaining case is one element is coming from C and two from B, but then the sum of positive numbers is at most m but each element coming from C is smaller than $-m$. Note that all operations take at most linear. ∎

→ Indeed we can also prove K-SUM and K-SUM′ are equivalent as well, but we leave it as an $\boxed{\text{exercise}}$.

Let's also define a geometric version of 3SUM.

GeomBase: given a set of n points with integer coordinate on three horizontal lines $y=0$, $y=1$ and $y=2$, determine whether there exists a non-horizontal line containing three of the points.

Thm: There is a subquadratic reduction from 3-SUM′ to GeomBase.

Pf: For each element $a \in A$ ($b \in B$) ($c \in C$) we create a point $(a,0)$ $(b,2)$ $(\frac{c}{2},1)$ on line $y=0$ ($y=2$) ($y=1$)

Line passing (x_1,y_1) (x_2,y_2): $\frac{y-y_1}{y_2-y_1} = \frac{x-x_1}{x_2-x_1}$

figure

It is immediate that three points $(a,0)$, $(b,2)$ and $(\frac{c}{2},1)$ are colinear iff $a+b=c$

Thm: There is a subquadratic reduction from GeomBase to 3-SUM′.

Pf: For each point $(a,0)$ create an element $a \in A$, for each point $(b,2)$ create an element $b \in B$ and for each point $(c,1)$ create an element $2c \in C$.

t's See another 3-Sum hard problem:

<u>3-Point-on-line</u>: Given a set of points in the plane, is there a line that contains at least three of the points.

<u>Thm</u>: There is a subquadratic reduction from 3-Sum to 3-point-on-line.

<u>Pf</u>: We are given a set S of n integers. for $x \in S$ we transform it to (x, x^3). Now we claim $a+b+c=0$ if and only if (a, a^3), (b, b^3) and (c, c^3) are collinear (which can be proved via elementary math and formula for line passing through two points and left as an $\boxed{\text{exercise}}$) ■

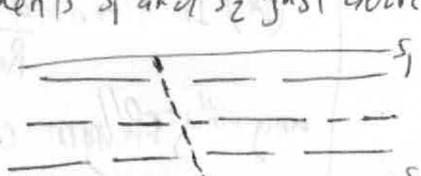
<u>Point-on-3-lines</u>: Given a set of lines in the plane, is there a point that lies on at least three of them.

Indeed point-on-3-lines and 3-point-on-line are dual (under transformations which makes a point a line and a line a point. So point-on-3-lines is 3-Sum hard as well.

<u>Visibility-Between-Segments</u>: Given a set S of n horizontal line segments in the plane and two particular horizontal segments s_1 & s_2, determine whether there are points does not intersect any segment in S.

<u>Thm</u>: There is a subquadratic reduction from GeomBase to Visibility-Between-Segments

<u>Pf</u>: let the x-coordinates of the points on the first line $A: y=0$ ordered from left to right be $a_1, a_2, \dots a_i$. Similarly let the points on the other lines $B: y=1$ and $C: y=2$ be $b_1 \dots b_j$ and $c_1 \dots c_k$. Let $\varepsilon = \frac{1}{5}$. Now transform the points on A into horizontal segments o A with x-intervals $[a_1+\varepsilon, a_2-\varepsilon] \dots [a_{i-1}+\varepsilon, a_i-\varepsilon]$. Similarly do this for lines B and C. Now place segments s_1 and s_2 just above and below the set of holes in A & C. Note the whole transformation can be done by sorts and thus needs $O(n \log n)$ time. It is obvious that if there is a line through points a on A, b on B and c on C, clearly s_1 can see s_2. For the reverse let L be the line which does not intersect any segments. Thus L should run through three "holes" $(a-\varepsilon, a+\varepsilon)$ on A, $(b-\varepsilon, b+\varepsilon)$ on B and $(c-\varepsilon, c+\varepsilon)$ on C.



Let the line intersects the holes at positions $(a+\delta_1)$, $(b+\delta_2)$ and $(c+\delta_3)$ so that $(a+\delta_1)+(b+\delta_2)=2(c+\delta_3)$ for $-\varepsilon<\delta_1,\delta_2,\delta_3<\varepsilon$. Because $\varepsilon=\frac{1}{5}$, $|\delta_1+\delta_2+2\delta_3|\leq\frac{4}{5}<1$ and since a,b,c are integers this is only possible when $a+b=2c$ (otherwise $\delta_1,\delta_2,\delta_3$ cannot compensate the difference), that is when there is a line through points A, B and C. ■

→ There are several other geometric problems which are 3-sum hard, e.g. given a set of triangles in the plane, does their union have a hole? or some other problems in motion planning: given a set of line segments, determine whether a line segment robot (a rod or ladder) can be moved (allowing both translation and rotation) from a given source to a given goal configuration without colliding with the obstacles (the best solution for this problem take $O(n^2)$ time [Vegter, Sirat '90]). You can find the details of these reductions in [Gajentaan and Overmars' 95].

Note that we can shave poly log factors for 3-sum, e.g. $O\left(\frac{n^2}{\frac{\log^2 n}{\log\log^2 n}}\right)$ by [Baran, Demaine and Patrascu' 08] via a randomized algorithm, or under some non-regular decision tree [Gronlund and Pettie' 14] we can even get $\tilde{O}(n^{1.5})$ but still 3-sum hardness conjecture holds on standard RAM model (✱)

You may also look at [Borassi, Crescenzi and Habib' 14] for some other problems which will be truly quadratic unless the well known strong Exponential Time Hypothesis (SETH) fails.
i.e., SAT does not have $O(2^{\delta n})$ time algorithm for $\delta<1$

(✱) In standard RAM model (Random Access Machine) with w-bit (w is often constant, 32 or 64), we have access to any cell of memory in constant time and all basic operations on words takes constant time

→ Note that we can have $O(n^2)$ algorithm for 3-SUM' (and hence 3-SUM because of equivalence) as follows: Sort B and C in $O(n\log n)$. Then for each $a\in A$, check whether $B+a$ (set of all number in B with a added to it) and C have any intersection in $O(n)$ by simultaneous tranversal of the ordered sets. Since $|A|=O(n)$, the running time is $O(n^2)$.