Convolution 3SUM

<u>Def.</u> (Convolution 3SUM)
Input: Array $A[0 \dots n-1]$ of integers
Task: Decide if there are i, j s.t. $A[i] + A[j] = A[i+j]$

Trivial Algorithm: $O(n^2)$ | Equivalent: $A[i] + B[j] = C[i+j]$

<u>Thm:</u> If there is an algorithm with running time $O(n^{2-\epsilon})$ for Convolution 3SUM, then there is an algorithm with running time $O(n^{2-\delta})$ for 3SUM.

<u>Proof:</u> Consider 3SUM instance $I \subseteq [1 \dots U]$
Preprocessing: Check if there is a 3SUM witness $x + x = z$
$O(n \log n)$ by sorting / binary search

Pick, uniformly at random, hash function h from almost balanced and <u>almost linear</u> family $h: [U] \to [R]$ (param. R)
For sake of demonstration: assume linear here
Apply hash function to each $a \in I$
(1) For each $x \in [R]$ s.t. $|h^{-1}[x]| > 3\frac{n}{R}$ ("heavy")
Check if there is a 3SUM witness $a+b=c$ for each $a \in h^{-1}(x)$
↳ For each $a \in h^{-1}(x)$ and $b \in I$ store $a+b$ in a set data structure
- For each $c \in I$, check if c contained in set data structure

→ Time $O(|h^{-1}(x)| \cdot n)$ hash set
$O(|h^{-1}(x)| \cdot n \log n)$ binary search tree

→ Time $\sum_{x: |h^{-1}(x)| > 3\frac{n}{R}} O(|h^{-1}(x)| \cdot n)$

$= O(R \cdot n)$ in expectation (almost balanced property)

(2) Assume that each $h^{-1}(x)$ $(x \in [R])$ contains $\le 3\frac{n}{R}$ elements

For all triples $i, j, k \in [3\frac{n}{R}]$

  Create array A of length $2 \cdot 8 \cdot R$

  For $x \in [R]$ offset $O := 8R$

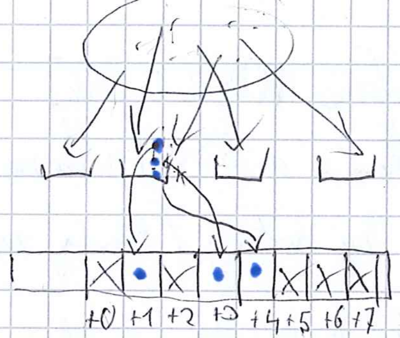
    Put i-th element of $h^{-1}(x)$ to $A[8(x-1)+1]$ and $A[8(x-1)+1+O]$

    Put j-th —"— $A[8(x-1)+3]$ and $A[8(x-1)+3+O]$

    Put k-th —"— $A[8(x-1)+4]$ and $A[8(x-1)+4+O]$

  Set all other entries to ∞ (or sufficiently large value)

Return yes if at least one Conv. 3SUM instance returns yes



<u>Correctness:</u> If $a=b$, then found in preprocessing. Otherwise

If $a+b=c$, then $h(a)+h(b)=h(c) \pmod R$ (linearity)

either $h(a)+h(b)=h(c)$ or $h(a)+h(b)=h(c)+R$

$\Rightarrow$ Let ~~i~~ a be i-th element of $h^{-1}(h(a))$, j, k accordingly

$\Rightarrow$ There is a triple i, j, k, s.t. in array A

$A[8(h(a)-1)+1] + A[8(h(b)-1)+3] = \cancel{A[8h(c)+4} \; A[8(h(c)-1)+4]$

or $A[8(h(a)-1)+1+O] + \ldots +O = \ldots +O$ ✓

If there is an instance A s.t. $A[i]+A[j]=A[i+j]$

Then we must have $i \underset{(\text{mod } 8)}{=} 8t_1+1$ and $j = 8t_2+3$

(because $x+y=z$ has unique solution over $\{1,3,4\}$ and $A[i] \neq A[j]$)

$\Rightarrow$ must find $a \in h^{-1}(t_1+1)$, $b \in h^{-1}(t_2+1)$, $c \in h^{-1}(t_1+t_2+1)$

s.t. $a+b=c$

<u>Running Time:</u> $O(n \log n + nR + (\frac{n}{R})^3 \cdot n^{2-\varepsilon})$

Set $R = n^{1-\varepsilon/4}$ $\rightarrow O(n^{2-\varepsilon/4})$

From Convolution 3SUM to 0-weight triangle

Definition: (0-weight-triangle)

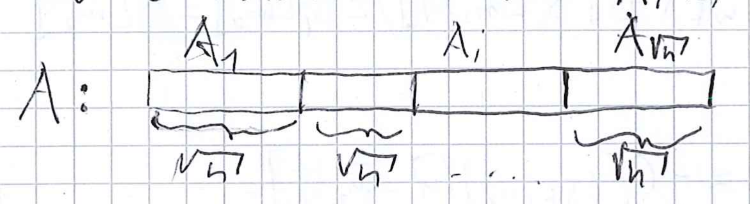
Input: Weighted directed graph G with positive/negative edge weights

Task: Decide if G contains a triangle of weight exactly 0

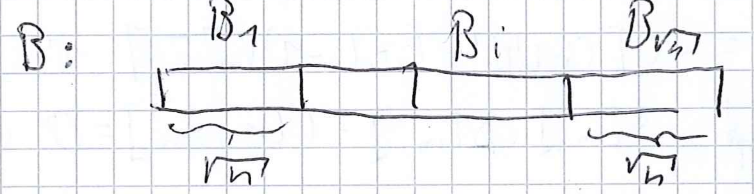
Theorem: If there is an algorithm for 0-weight-triangle on graphs with n nodes with running time $O(n^{3-\varepsilon})$ (for some constant $\varepsilon > 0$), then there is an algorithm for Convolution 3SUM on arrays of length n with running time $O(n^{2-\delta})$ (for some constant $\delta > 0$)

Proof:

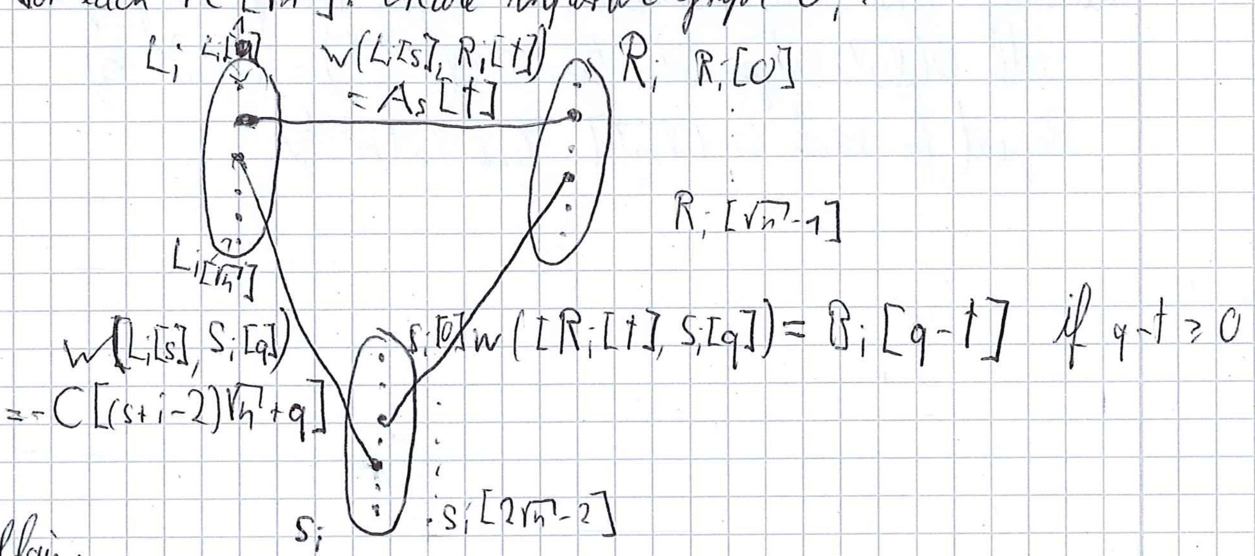
Consider Convolution 3SUM instance A, B, C



$A_i[j] = A[(i-1)\lceil\sqrt{n}\rceil + j]$



For each $i \in [\lceil\sqrt{n}\rceil]$: create tripartite graph G_i:



Claim:

There is a 0-weight triangle $L_i[s], R_i[t], S_i[q]$ for some $i \in \lceil\sqrt{n}\rceil$

$\iff$ there are m_1, m_2, m_3 s.t. $A[m_1] + B[m_2] = C[m_1 + m_2]$

BRUNNEN

⇒ 0-weight triangle

$\Rightarrow A_s[t] + B_i[q-t] - C[(s+i-2)\sqrt{n}+q] = 0$

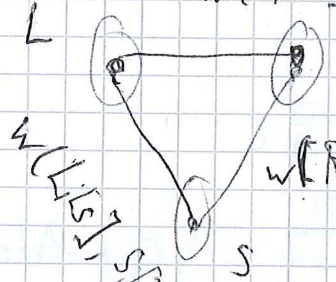
$\Leftrightarrow A[\underbrace{(s-1)\sqrt{n}+t}_{=:m_1}] + B[\underbrace{(i-1)\sqrt{n}+q-t}_{=:m_2}] = C[\underbrace{(s+i-2)\sqrt{n}+q}_{=m_1+m_2}]$ ✓

⇐ Consider some m_1, m_2 s.t. $A[m_1] + B[m_2] = C[m_1+m_2]$

Write $m_2 = (i-1)\sqrt{n} + \bar{m}_2$ for some $\bar{m}_2 \in \{0, \dots, \sqrt{n}-1\}$, $i \in \{1, \dots, \sqrt{n}+1\}$

$m_1 = (s-1)\sqrt{n} + t$ for some $t \in \{0, \dots, \sqrt{n}-1\}$, $s \in \{1, \dots, \sqrt{n}+1\}$

Consider G_i: $w(L_i[s], R_i[t]) = A_s[t] = A[m_1]$



$w(R_i[t], S_i[\bar{m}_2+t]) = B_i[\bar{m}_2] = B[m_2]$

$w(L_i[s], S_i[\bar{m}_2+t]) = -C[(s+i-2)\sqrt{n}+\bar{m}_2+t] =$

$= -C[(s-1)\sqrt{n}+t+(i-1)\sqrt{n}+\bar{m}_2] = -C[m_1+m_2]$ ✓

Weight of triangle: $A[m_1] + B[m_2] - C[m_1+m_2] = 0$ ✓

Running time: By assumption, 0-weight triangle on graph with $O(\sqrt{n})$ edges takes time $O(\sqrt{n}^{3-\varepsilon}) = O(n^{3/2-\varepsilon/2})$

Repeat for each $i \in [\sqrt{n}]$ time $O(n^{2-\frac{\varepsilon}{2}})$