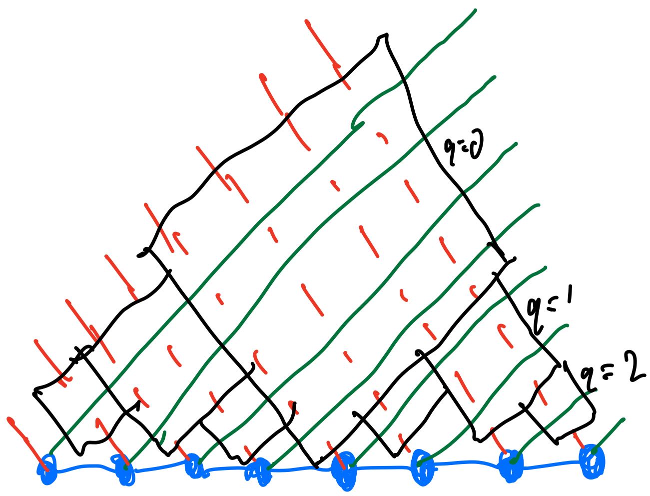
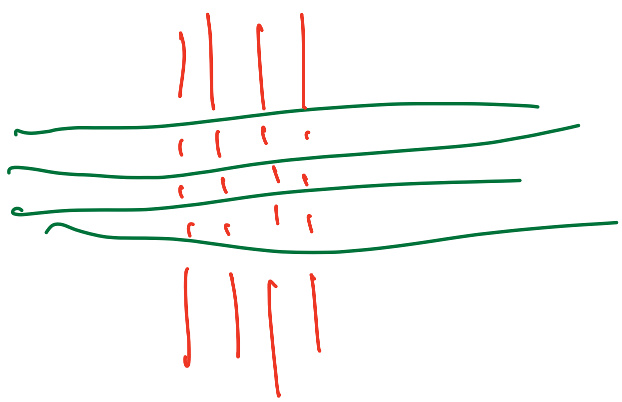




Problem.

$$\underbrace{\#\left\{ (b_i) \in \prod_{i=1}^{2d} B_i : \begin{array}{l} t(b_1) < \cdots < t(b_{2d}) \\ \forall j \in \{1, \ldots, d\} : z(b_{\alpha(j)}) < z(b_{\beta(j)}) \end{array} \right\}}_{A}$$

$$C_j = \{(b,b') \in B_{\alpha(j)} \times B_{\beta(j)} : z(b) < z(b')\}$$

(set of crossings) $A(C_1, \dots, C_d)$

$$A \cong \{(c_j) \in \prod_{j=1}^{d} C_j : t(c_{\gamma(1)}) < \cdots < t(c_{\gamma(d)})\}$$

$$\gamma(i) = \begin{cases} (j,1) & \alpha(j) = i \\ (j,2) & \beta(j) = i \end{cases}$$

$$c_{\gamma(i)} = (c_j)_k \in B_i \qquad \gamma(i) = k$$

$$\delta : \{1, \dots, d\} \times \{1,2\} \xrightarrow{\sim} \{1, \dots, 2d\} \qquad (\alpha + \beta)$$

$$\delta(j,1) = \alpha(j) \qquad \text{(bijection)}$$

$$\delta(j,2) = \beta(j)$$

$$\gamma = \delta^{-1}$$

$$C_j = \bigcup_{0 \le q < p} C_{j,q} \qquad C_{j,q} = \bigcup_{|\sigma| = q} C_{j,\sigma}$$

$$C_{j,\sigma} = B_{j,\sigma,0} \times B_{j,\sigma,1}$$

$$B_{j,\sigma,0} = \{ b \in B_{\alpha(j)} : z(b) = \sigma 0 \cancel{*} \}$$

$$B_{j,\sigma,1} = \{ b \in B_{\beta(j)} : z(b) = \underbrace{\sigma}_{q-1} \underbrace{(*)}_{p-q-1} \}$$

$C_{j,q}$ is a union of 2^q squares w/ side-length $\leq L/2^q$

"$C_{j,q}$ has perimeter L"

$$A(C_1, \dots, C_d) = \bigcup_{\bar{q} \in \{0,\dots,p-1\}^d} \overbrace{A(C_{1,q_1}, \dots C_{d,q_d})}^{A_{\bar{q}}}$$

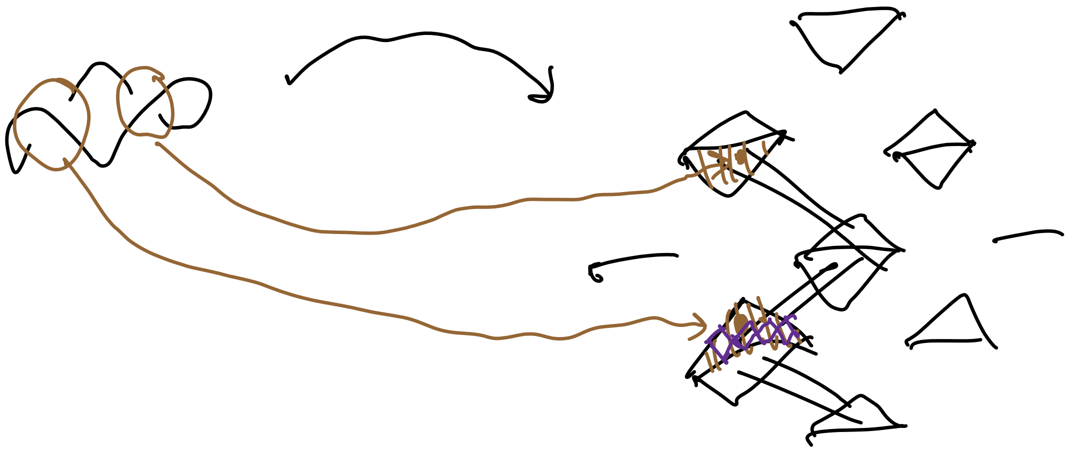
Subproblem: Count $A_{\bar{q}}$

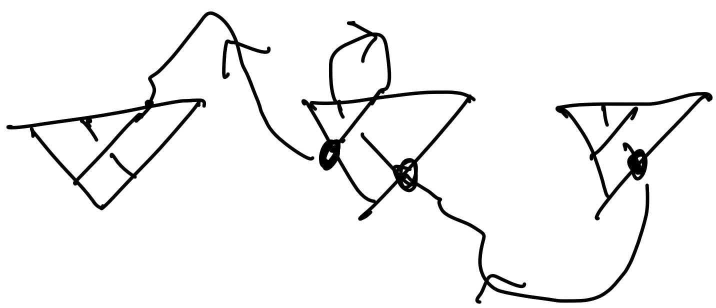
Algorithm 1: $A_{\bar{q}} = \bigcup_{\substack{\bar{\sigma} \\ |\sigma_j| = q_j}} A(C_{j,\sigma_j})$

Each $|A(C_{j,\sigma_j})|$ is calculated w/ Lem. 4.1

Time $2^{\Sigma q_i} \cdot \underbrace{\max(L/2^{q_i})}_{L/2^{\min(\bar{q})}} = L \cdot 2^{\Sigma q_j - \min(\bar{q})}$

Count embeddings





Algorithm 2: 2D algorithm

$C = \{\text{crossings}\}$

2D algorithm calculates

$$|A(C, C, \ldots, C)|$$

It can be adapted to calculate

$$|A(C_1, \ldots, C_d)|$$

This has time

$$(\max |C_j|)^{\frac{3}{4}d} = L^{2 \cdot \frac{3}{4}d} / 2^{\frac{3}{4}d \cdot \min(\bar{q})}$$

$$|C_{j,q}| \leq 2^q \cdot (L/2^q)^2 = L^2/2^q$$

Optimistically, this can be improved to

$$L^{\frac{3}{4}d} \Big/ \underbrace{\left(2^{\sum q_i - \min(\bar{q})}\right)^{3/4}}_{2^{\frac{3}{4}(d-1)q} \sim 2^{\frac{3}{4}d \cdot q}} = \left(\frac{\prod |c_i|}{\max |c_j|}\right)^{3/4}$$

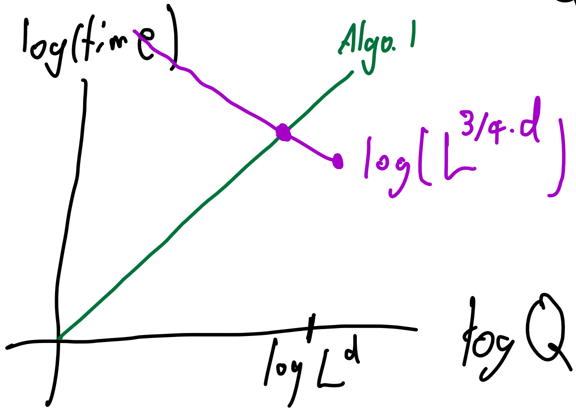
$q_i = q$

(d large)

$$Q := 2^{\sum q_i - \min(\bar{q})} \qquad 1 \le Q \le L^{d-1}$$

Algorithm 1: Time $L \cdot Q \sim Q$

Algorithm 2: Time $(L^{2d}/Q)^{3/4}$



Worst case $Q = \left(L^{2d}/Q\right)^{3/4}$

$$= L^{\frac{3}{2}d}/Q^{3/4}$$

$$Q^{7/4} = L^{\frac{3}{2}d}$$

$$Q = L^{\frac{4}{7}\cdot\frac{3}{2}d} = L^{\frac{6}{7}d} \text{ time}$$

for one choice of crossing fields

Overall time $L^{2d} \cdot L^{6/7 \cdot d} = L^{d20/7} = V^{\frac{20}{21}d}$

Improvement: Consider all crossing fields at once.

$$C_j = \left\{(b, b') : \begin{array}{l} \text{For some crossing field } F_x, \\ b, b' \text{ are in } F_x, \text{ and } z(b) < z(b') \end{array}\right\}$$

in right color, orient.

$$C_j = \bigcup C_{j,q} \qquad C_{j,q} = \bigcup_{\substack{C' \subseteq C_{j,q} \\ C' \text{ part. crossing field}}} C'_q$$

$C_{j,q}$ = "squares of size $2^{p \cdot q}$ in every crossing field at once"

Now: $C_{j,q}$ is a union of $L^2 \cdot 2^q$ squares of size $\le 2^{p-1-q} \sim L/2^q$, $|C_{j,q}| = L^4/2^q$

$2^p \sim L$

Algorithm 1: $\prod_{j=1}^{d} (L^2 \cdot 2^{q_j}) \cdot \frac{L}{2^{\min(\bar{q})}}$

$= L^{2d+1} \cdot Q \sim L^{2d} \cdot Q$

Algorithm 2: $\left(\frac{\prod |C_j|}{\max |C_j|} \right)^{3/4}$

$\sim \left(\frac{L^{4d}}{Q} \right)^{3/4} = L^{3d} / Q^{3/4}$

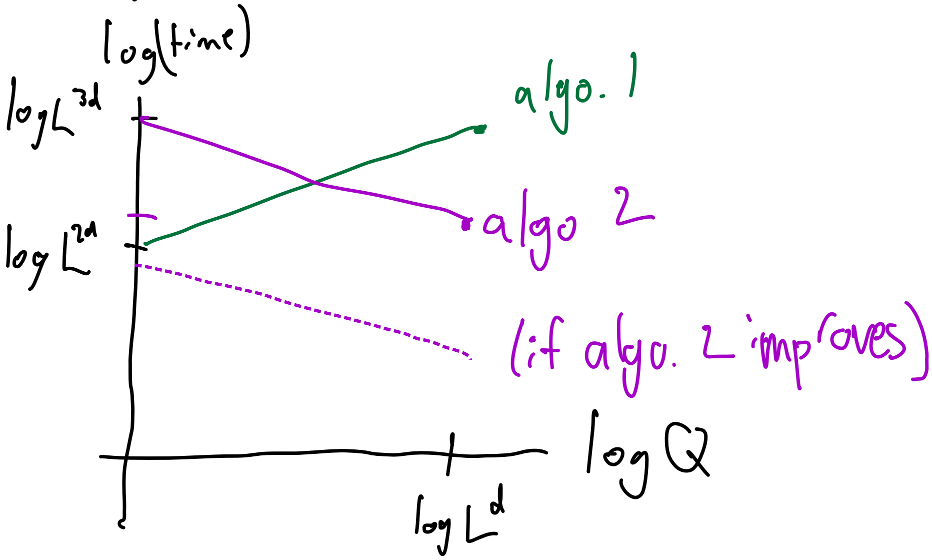
$\underline{L^{2d} \cdot Q} = \underline{L^{3d} / Q^{3/4}}$

$Q^{7/4} = L^d$

$Q = L^{4d/7}$

Worst case time $= L^{2d} \cdot Q = L^{18/7 d} = V^{\frac{18}{21} \cdot d} =$

$= V^{\frac{6}{7} d}$



At $Q = 1$:

Algo. 1: L^{2d} time

Algo. 2: $|C_j| = L^{4d}$

If there is a 2D algo. with time n^{ω}, then this is time

$$(L^{4d})^{\omega} = L^{4\omega d}$$

If $\omega \leq \frac{1}{2}$, then $L^{4\omega d} \leq L^{2d}$, so

Algorithm 2 dominates, and this is ~~not~~ a nontrivial 3D algorithm.

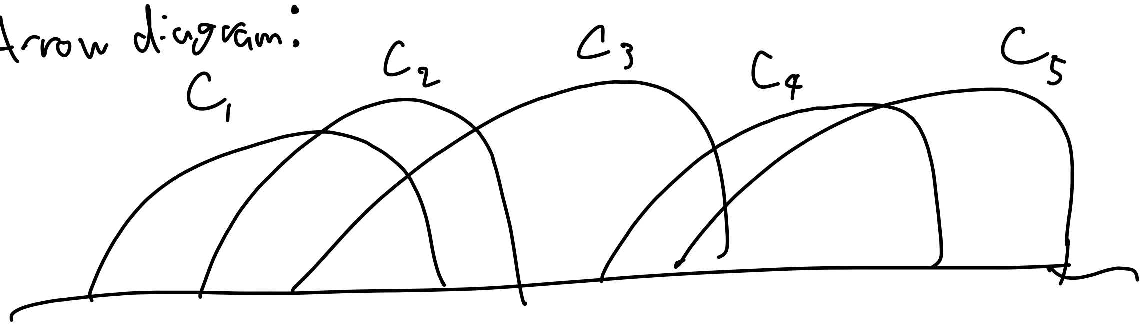
Tues. 12:30-15:00

Algorithm 2: Given $C_i \subseteq [0,L]^2$, calculate

$$|A(C_1,\dots,C_d)| = \left|\left\{(c_j) \in \prod_{j=1}^{d} C_j : t((c_{\gamma_0(1)})_{\gamma_1(1)}) < t((c_{\gamma_0(2)})_{\gamma_1(2)}) < \cdots < t((c_{\gamma_0(2d)})_{\gamma_1(2d)})\right\}\right|$$

$$\gamma : \{1,\dots,2d\} \to \{1,\dots,d\} \times \{1,2\}$$

Arrow diagram:

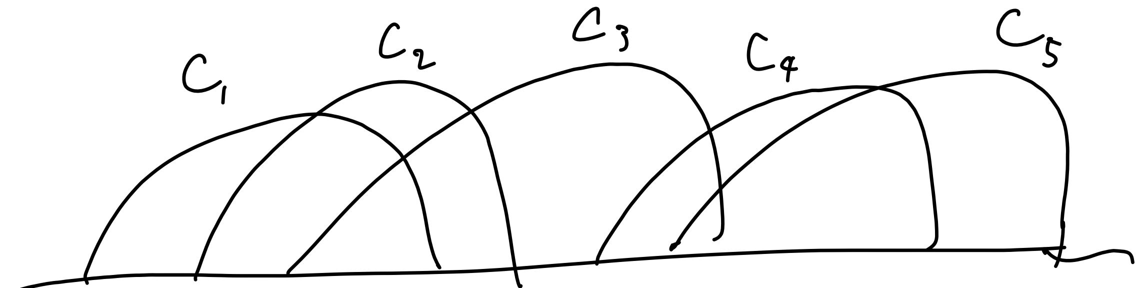


Naive algorithm: Search over all $\prod_{j=1}^{d} C_j$ and count

Run time: $\left|\prod_{j=1}^{d} C_j\right| = \prod_{j=1}^{d} |C_j|$

Almost-naive algorithm:

Goal: Calculate in time $\frac{\prod |C_j|}{\max |C_j|}$



Nevermind this,

Full problem: Count # times a Gauss diagram appears in a 3D knot.

$C := \{$crossings of 3D knot$\}$

(Ignoring color, ori, ...) we can to count

$|A(C, \ldots, C)|$ of 3D knot ✓

$C = \bigcup_{q=0}^{p-1} C_q$, $C_q := \{$crossings inside a $2^{p \cdot l \cdot l} \times 2^{p \cdot l \cdot l}$ square of a crossing field$\}$

$$A = \bigcup_{\bar{q} \in \{0,\dots,p-1\}^d} A_{\bar{q}}, \quad A_{\bar{q}} := A(C_{q_1},\dots,C_{q_d})$$

Subproblem: Calculate $|A_q|$ for a given $q \in \{0,\dots,p-1\}^d$.

$$Q_{old} = 2^{\left(\sum q_i\right) - \min(\bar{q})}$$

$$Q_{new} = 2^{\sum q_i}$$

$$L^{-1} \leq \frac{Q_{new}}{Q_{old}} \leq 1$$

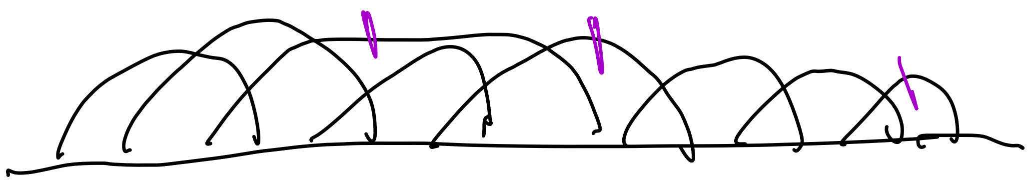
$L \ll L^d$, so we don't care about factors of L.

Algorithm 1/3: Time $L \cdot Q_{old} = Q \cdot L^{O(1)}$

Algorithm 2: Time $\left(\prod |C_i|\right)^{3/4} = \left(\prod \left(L^4/2^{q_i}\right)\right)^{3/4} =$

$= \left(L^{4d}/Q\right)^{3/4}$

Further improvement for Algorithm 2



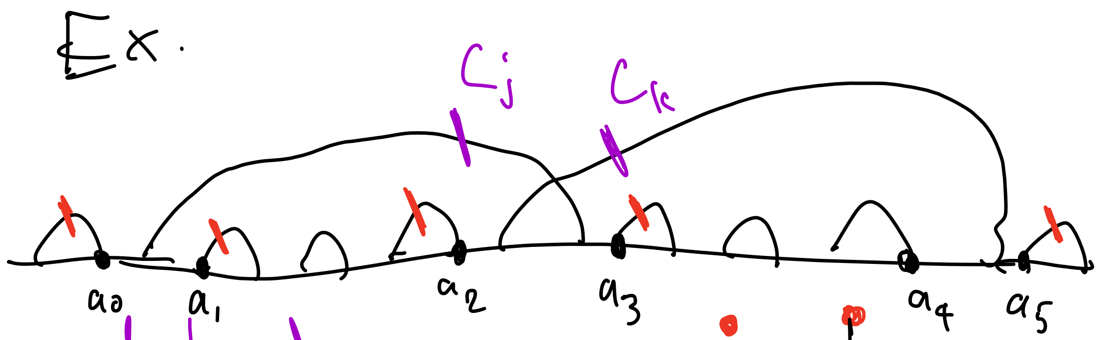
We pick crossings to delay which we can count with a lookup table.

Constraint: Two ends of delayed crossing cannot be adjacent.

We want to weaken this constraint.

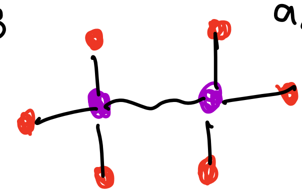
weaker constraint: The graph of adjacencies of delayed crossing is made of small connected components.

Ex.



• delayed

• not delayed



We need a lookup table to count all the crossings in a component at once.

For a choice of non-delayed crossings, we need to count

$$f(a_0, a_1, a_2, a_3, a_4, a_5) =$$

$$\Big|\{(w,x) \in C_j,\ (y,z) \in C_k :\ t(a_0) < t(w) < t(a_1),\ t(a_2) < t(y) < t(x) < t(a_3),\ t(a_4) < t(z) < t(a_5)\}\Big|$$

choices for $a_0 \leq \max |C_j| \leq L^4$

$|\mathrm{dom}\, f| \leq L^{4 \cdot 6} \leq L^{4 \cdot 3 \cdot |\text{component}|}$

Creating a lookup table:

For each $\bar{a} \in \mathrm{dom}\, f$:

Count over $C_j \times C_k$ how many satisfy the condition.

(time $|C_j| \cdot |C_k| \leq L^{4 \cdot |\text{component}|}$)

Overall time: $L^{12 \cdot |\text{component}|} \cdot L^{4 \cdot |\text{component}|}$

$= L^{16 \cdot |\text{component}|}$

How to pick delayed crossing:

For $p = \frac{1}{3} - \varepsilon$ we delay each crossing w/ independent probability p.

Then with prob. $\geq \frac{1}{2}$

$$\prod_{\text{non-delayed } j} |C_j| \leq \left(\prod_j |C_j|\right)^{\frac{2}{3}+\varepsilon}$$

Next, need to show delayed crossings are a graph with small components.

For a given crossing C, calculate the expected size of component containing C.

Fri. 2pm