$\Rightarrow$ Space complexity $2^{O(n)}$
time complexity $2^{O(n)}$ (Space^2).
$\hookrightarrow$ may be improved with better data structure for S

<u>What about correctness</u>? [Becker, Ducas, Gama, Laarhoven '16: $S = 2^{0.292 \cdot n}$]
[in practice: $n \approx 160$, Albrecht, Ducas, H, K, Postlethwaite, Stev. '19: $S = 2^{0.21 n}$]

Why don't we miss all $s \in L$ with $\|s\| = \lambda_1$.

Idea: perturb the t_i's to hide them to the algorithm.

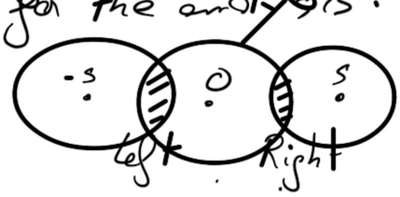
- Sample e_i uniformly in $\mathcal{B}(0, \xi \cdot \lambda_1)$ with $\xi > 1/2$. eg $\xi = 0.51$
- then write "$e_i \bmod \mathcal{B}$" as $\underset{\text{small}}{e_i} + \underset{L}{b_i} = \tilde{b}_i$ $\hookrightarrow$ close to L.

When sieving, sieve with $\tilde{b}_i$, but add b_i to T.
if $b_j \in T$ with $\|\tilde{b}_i - b_j\| \leq (1-\varepsilon)\|\tilde{b}_i\|$
then $b_i \leftarrow b_i - b_j$
$\tilde{b}_i \leftarrow \tilde{b}_i - b_j = b_i^{(new)} + e_i$

Alg.

At the end, output $b_i - b_j$ for $b_i, b_j \in T$ closest.
* The space complexity bound of $2^{O(n)}$ still holds
Proof of <u>correctness</u>: consider any the $(b_i, \tilde{b}_i)$ s.t. $\tilde{b}_i - b_i$ belongs to $\mathcal{B}(0, \xi \cdot \lambda) \cap \mathcal{B}(s, \xi \cdot \lambda)$.
$\hookrightarrow$ s arbitrary reaching λ_1.

We analyze the algorithm, using another one. The latter is not efficient, but is used only for the analysis.



At start, we change e_i to e_i' as follows:

if $e_i \notin$ Left $\cup$ Right, $e_i' := e_i$ (1)

τ: if $e_i \in$ Left, $e_i' := e_i + s$ or keep* (2)

if $e_i \in$ Right, $e_i' := e_i - s$ or keep* (3)

*with prob. 1/2.

The $\tilde{b}_i$'s are the same, only some underlying b_i's do change. In case (2), b_i may be mapped to $b_i' := b_i + s$, in (3), it may be mapped to $b_i' := b_i - s$

If A' outputs $b_i - b_j = 0$ for some switched $b_i' \in$ (2) and unswitched $b_j' \in$ (2) then A would output $b_i - b_j = s$.

$\Rightarrow$ it suffices to bound from below the probability that A' outputs $b_i' - b_j' = 0$ for $b_i' \in$ (2) switched and $b_j' \in$ (2) unswitched.

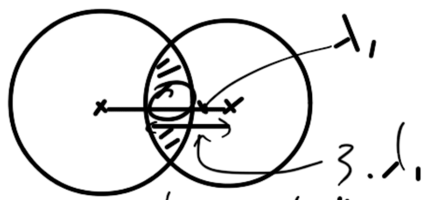
As τ maps $U(\mathcal{B}(0, 3\lambda_1) \bmod B)$ to itself, up to the final step, A and A' are the same!

$\Rightarrow$ it suffices to do it for A, and divide the success prob by 1/4 (correct switches)

We initially sample $N = \frac{\{\text{Space bound}\}}{\Pr[(2)]} \cdot 4$ vectors.

There are $\geq 4 \cdot \{\text{Space bound}\}$ of them that belong to (2).
By pigeon-hole principle, there are two that collide :-)

* What is $\Pr[(2)]$?



$\Rightarrow$ contains a ball of radius $\geq \lambda_1 \cdot (3 - \frac{1}{2})$

$\Rightarrow$ prob $\geq (3 - \frac{1}{2})^{n}$.

* in the algorithm, we need to know λ_1 !

Try $\bar{\lambda}_1 = 1, 1+\varepsilon, (1+\varepsilon)^2, \ldots$ and run the algorithm for each guess!

Best provable time bound with this approach:
Pujol-S '09 $2^{2.465n+o(n)}$

Best provable time bound for SVP: $2^{n+o(n)}$
Aggarwal, Dadush, Regev, Stephens-Davidowitz '15.

More interesting question: understand and/or improve the best algorithms based on heuristics.