

Sieving for shortest vectors in lattices using angular locality-sensitive hashing and quantum search

Thijs Laarhoven, Michele Mosca, Joop van de Pol

mail@thijs.com
<http://www.thijs.com/>

PQCrypto 2014, Waterloo, Ontario, Canada
(October 3, 2014)

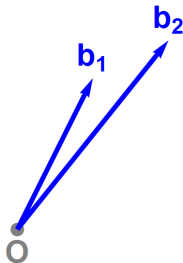
Lattices

What is a lattice?



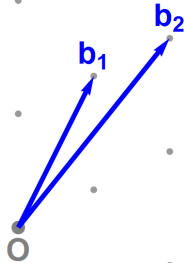
Lattices

What is a lattice?



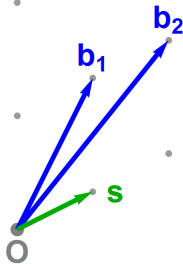
Lattices

What is a lattice?



Lattices

Shortest Vector Problem (SVP)



Sieving

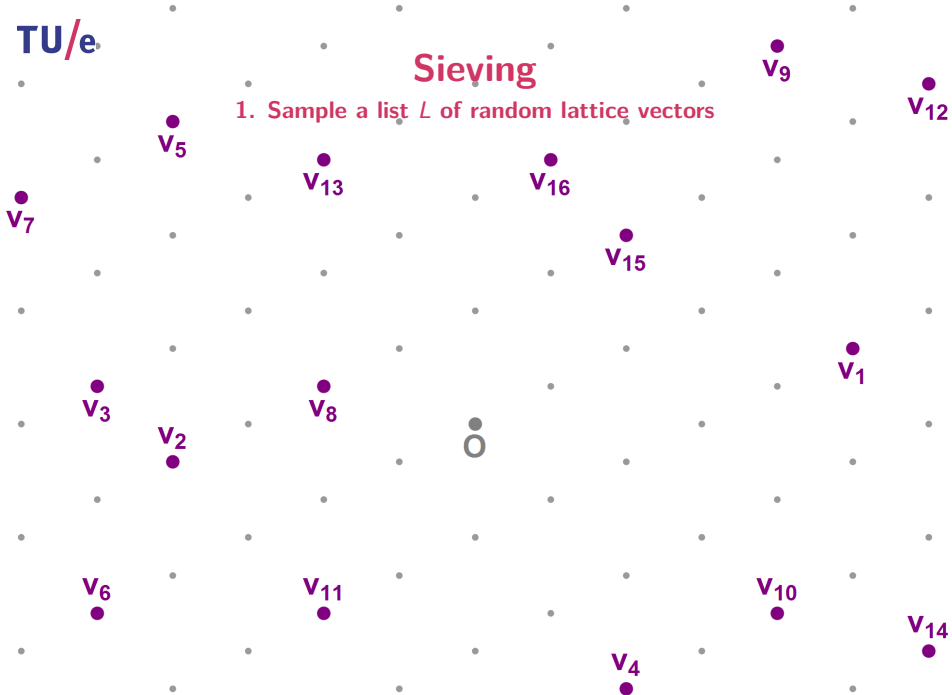
1. Sample a list L of random lattice vectors



A 2D lattice of points is shown, with the origin labeled O . The lattice points are arranged in a regular grid pattern. The origin O is marked with a larger dot and the letter O below it.

Sieving

1. Sample a list L of random lattice vectors



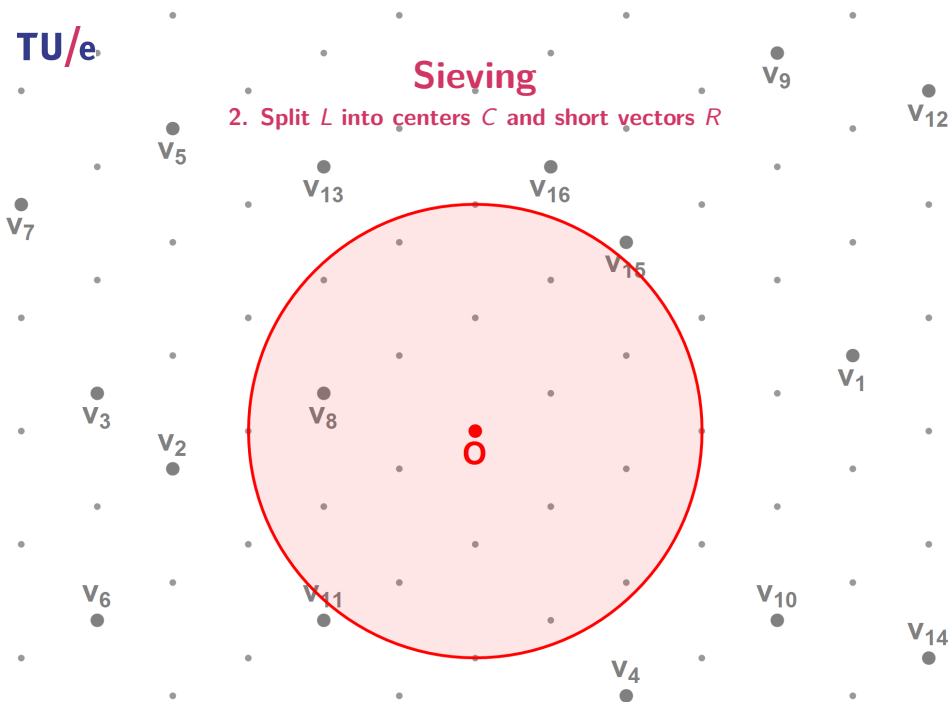
Sieving

2. Split L into centers C and short vectors R



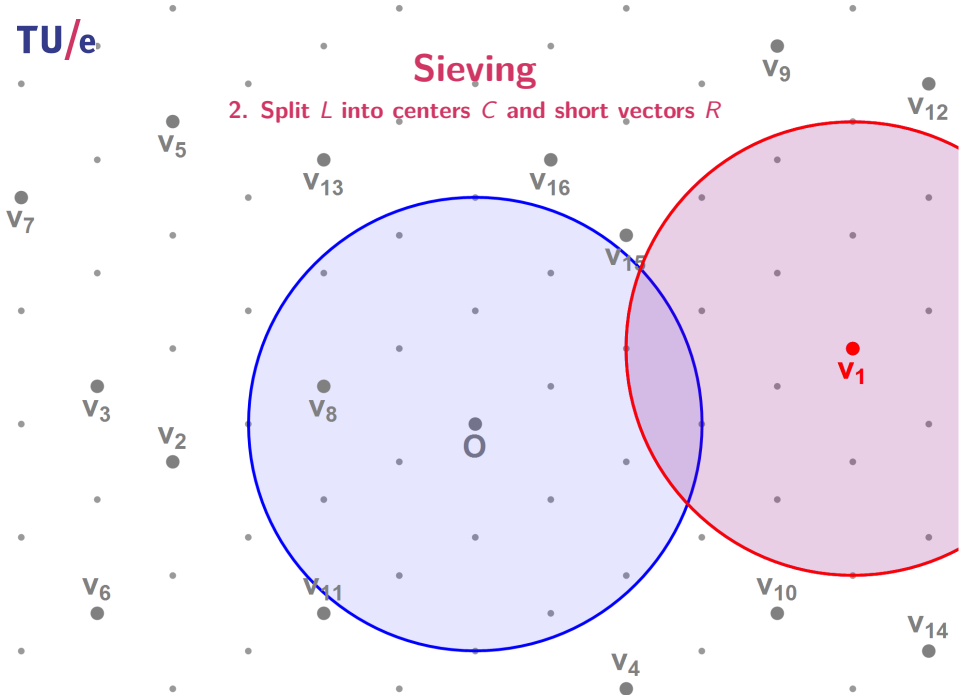
Sieving

2. Split L into centers C and short vectors R



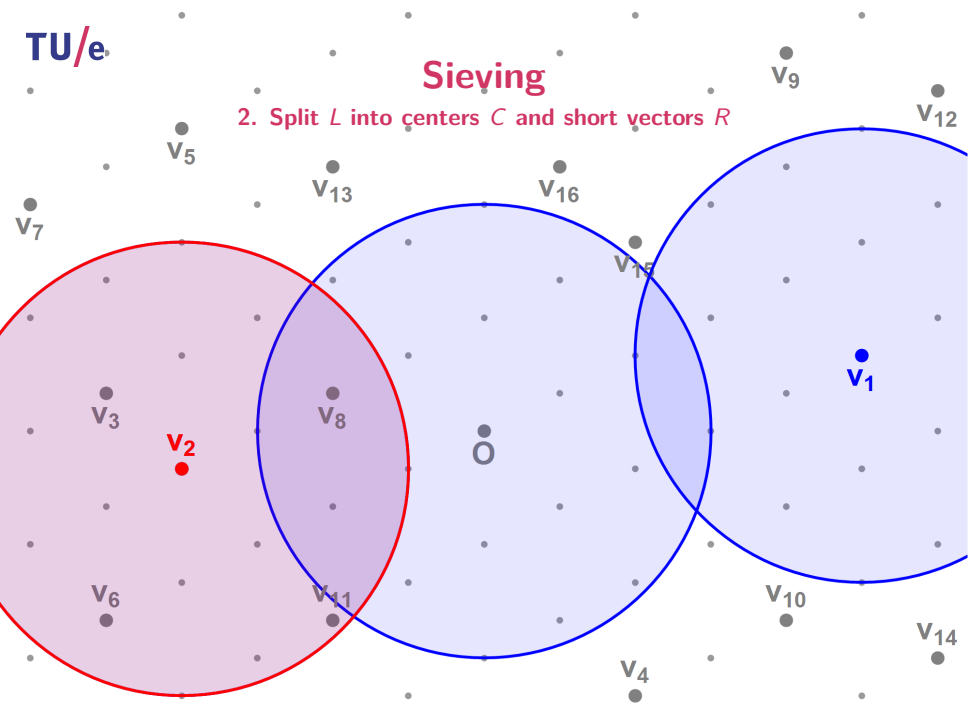
Sieving

2. Split L into centers C and short vectors R



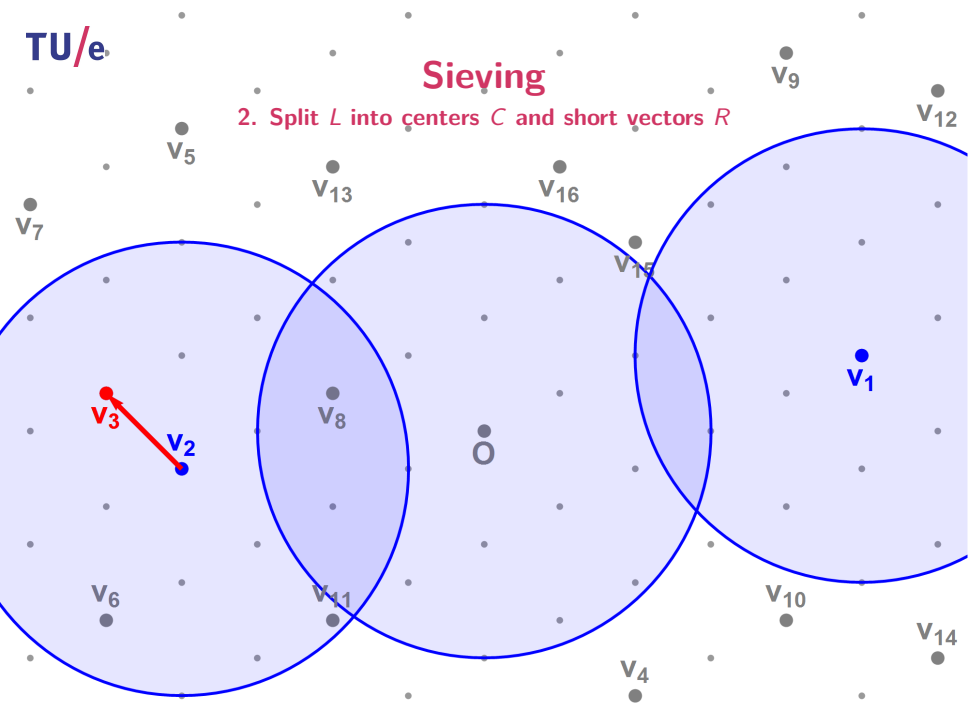
Sieving

2. Split L into centers C and short vectors R



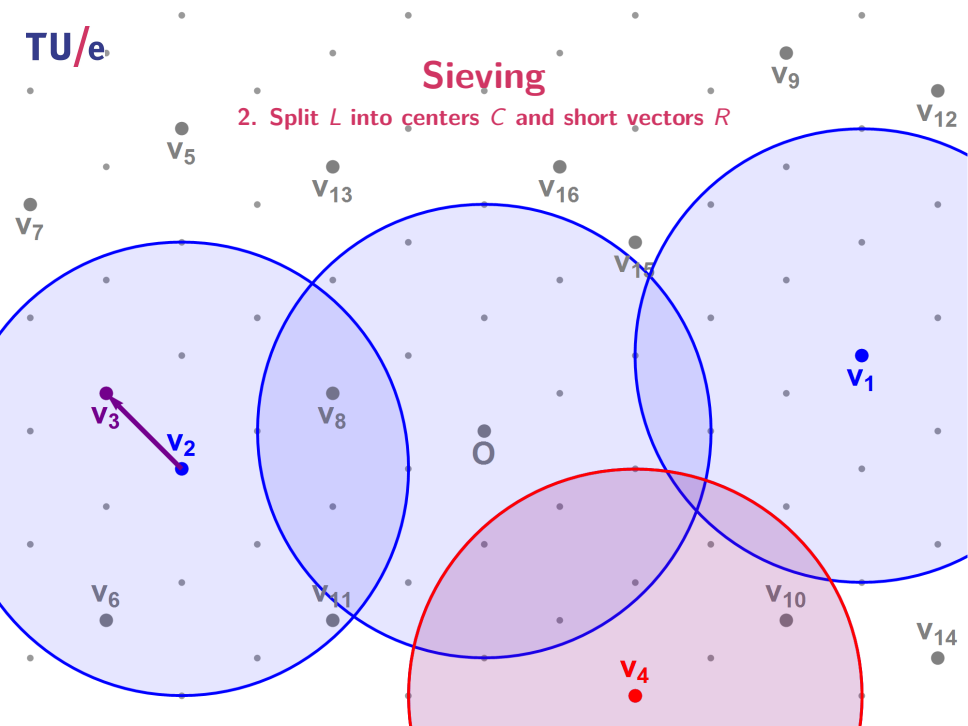
Sieving

2. Split L into centers C and short vectors R



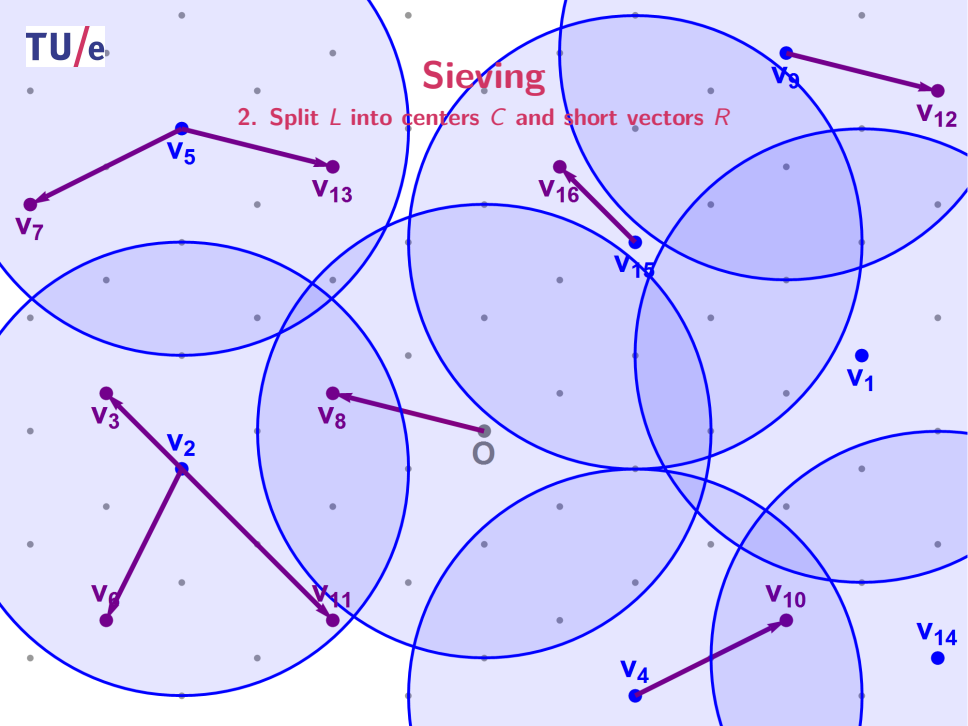
Sieving

2. Split L into centers C and short vectors R

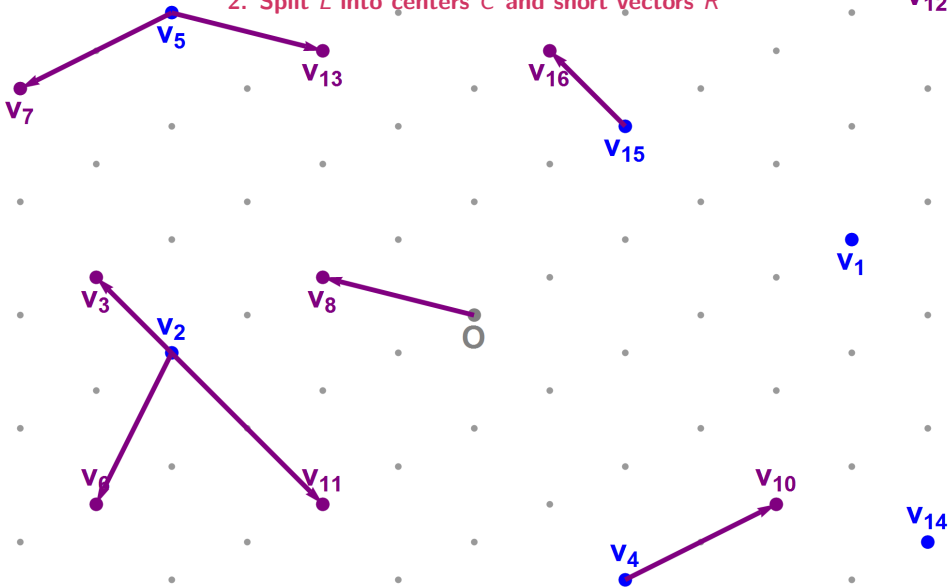


Sieving

2. Split L into centers C and short vectors R

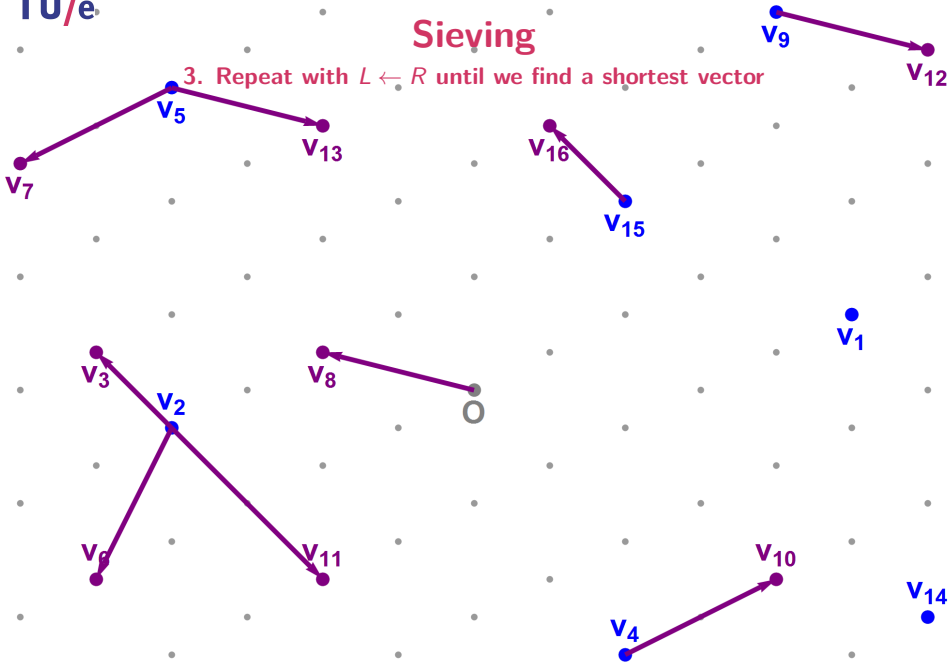


Sieving

2. Split L into centers C and short vectors R 

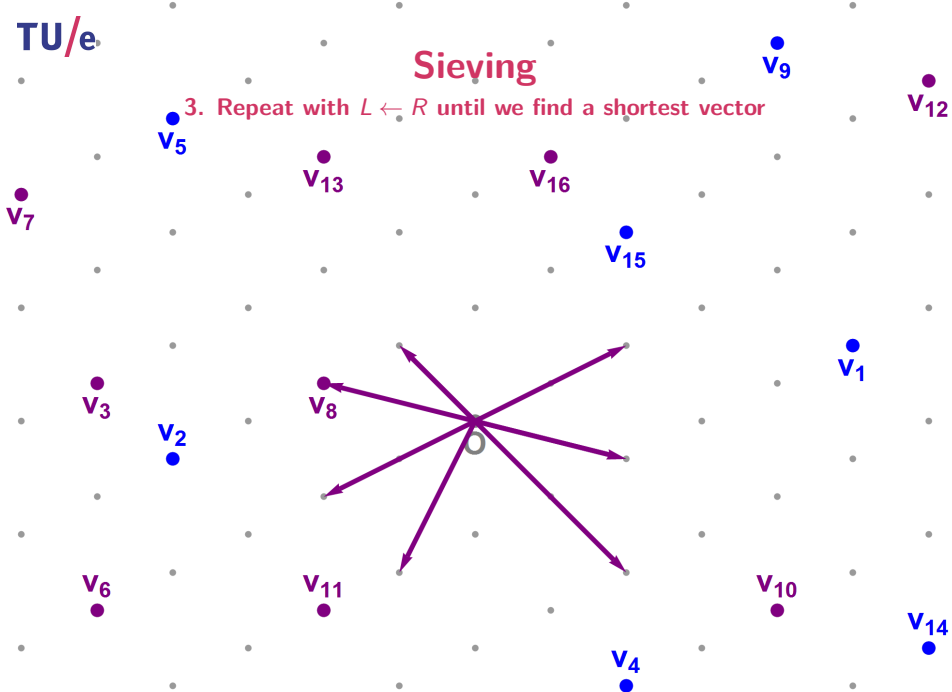
Sieving

3. Repeat with $L \leftarrow R$ until we find a shortest vector



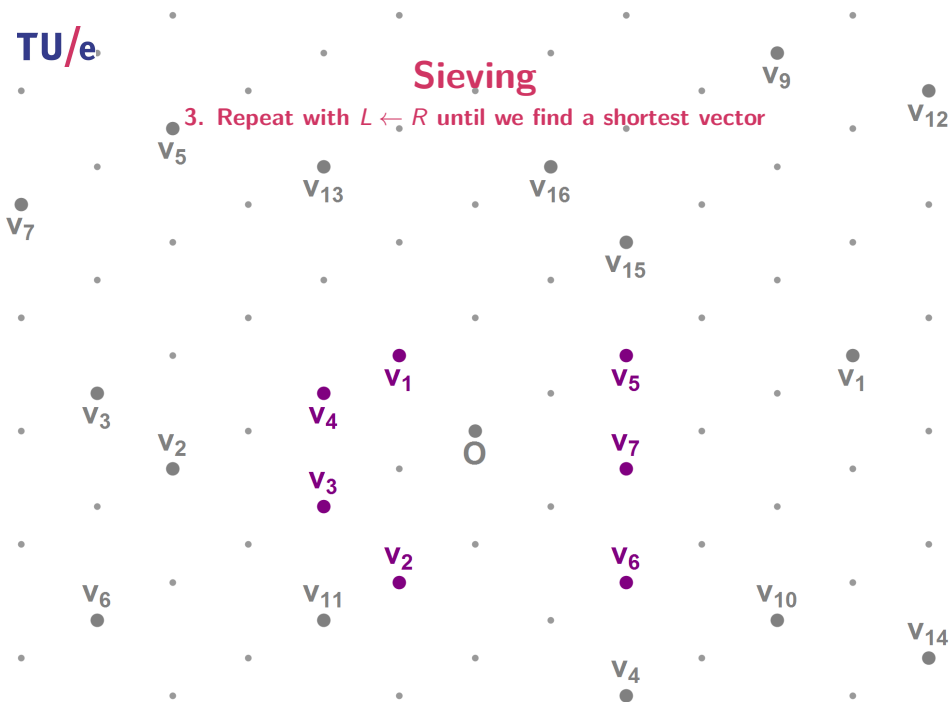
Sieving

3. Repeat with $L \leftarrow R$ until we find a shortest vector



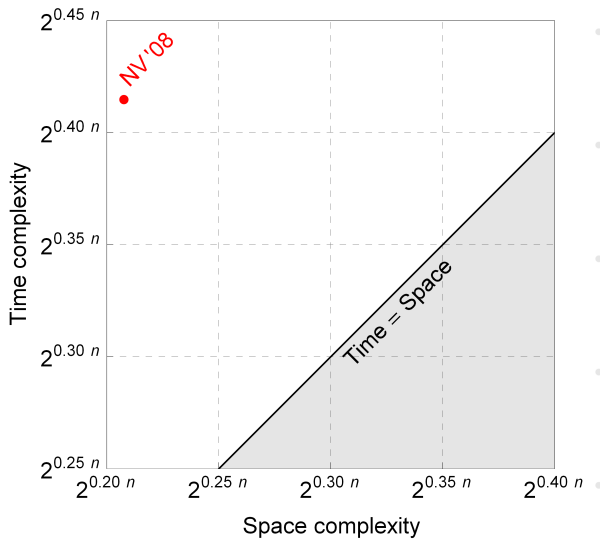
Sieving

3. Repeat with $L \leftarrow R$ until we find a shortest vector



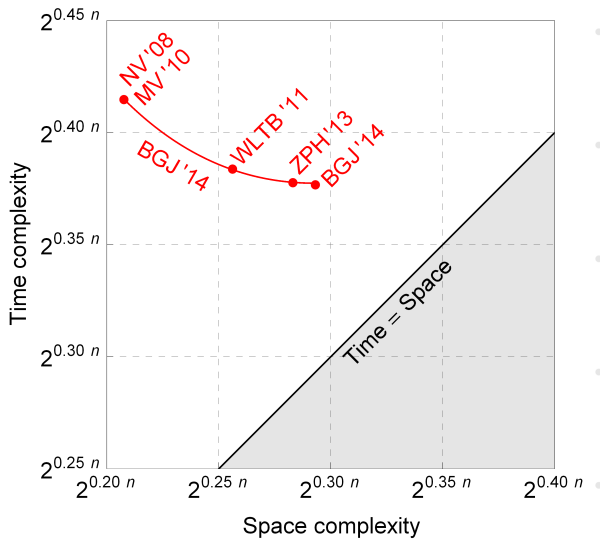
Sieving

Space/time trade-off



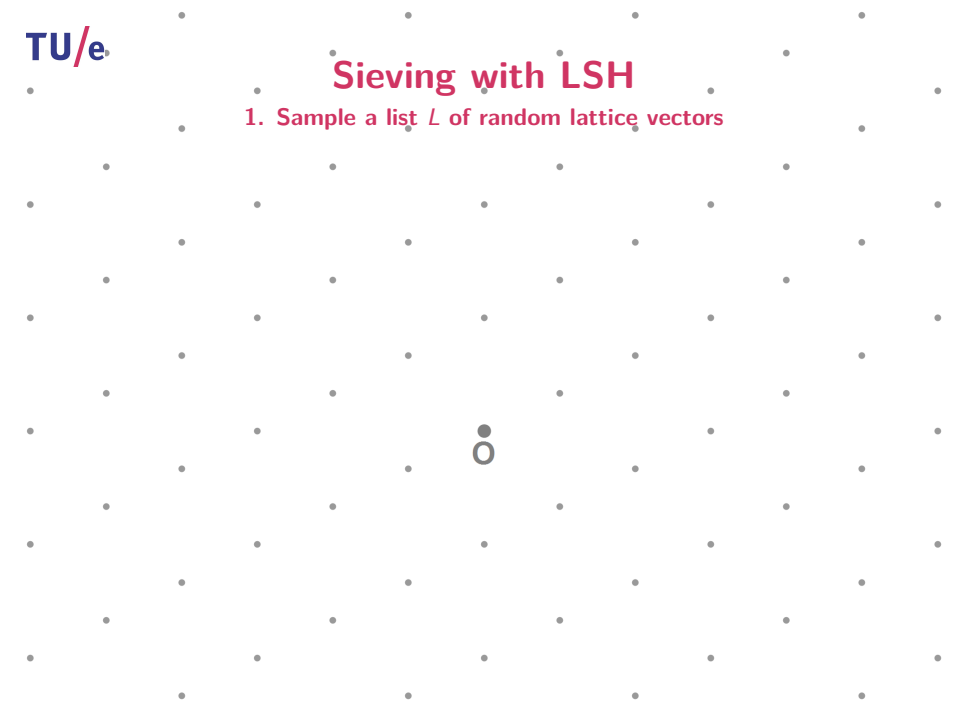
Sieving

Space/time trade-off



Sieving with LSH

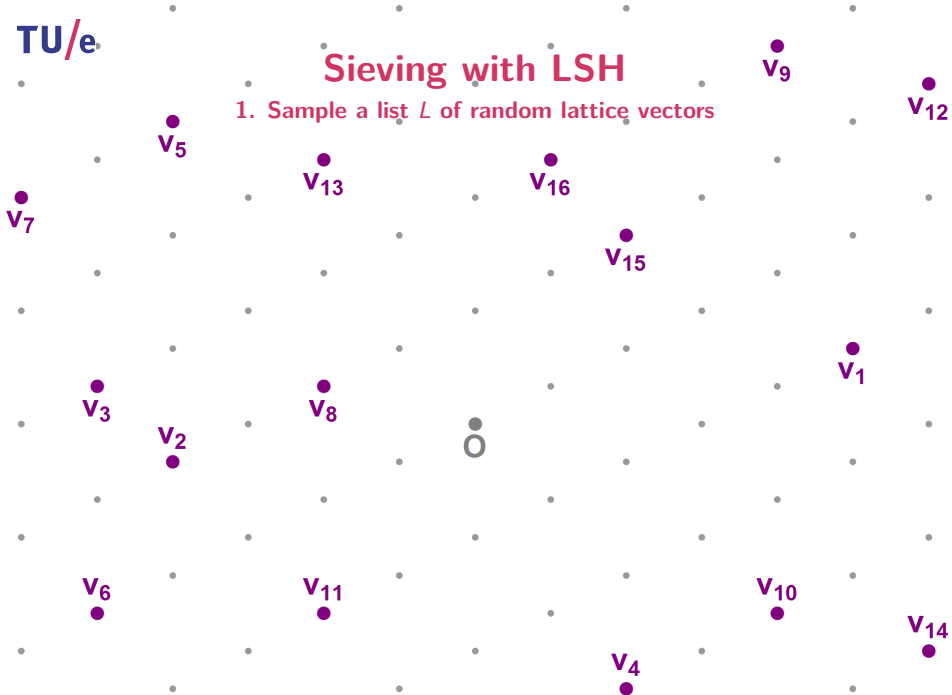
1. Sample a list L of random lattice vectors



O

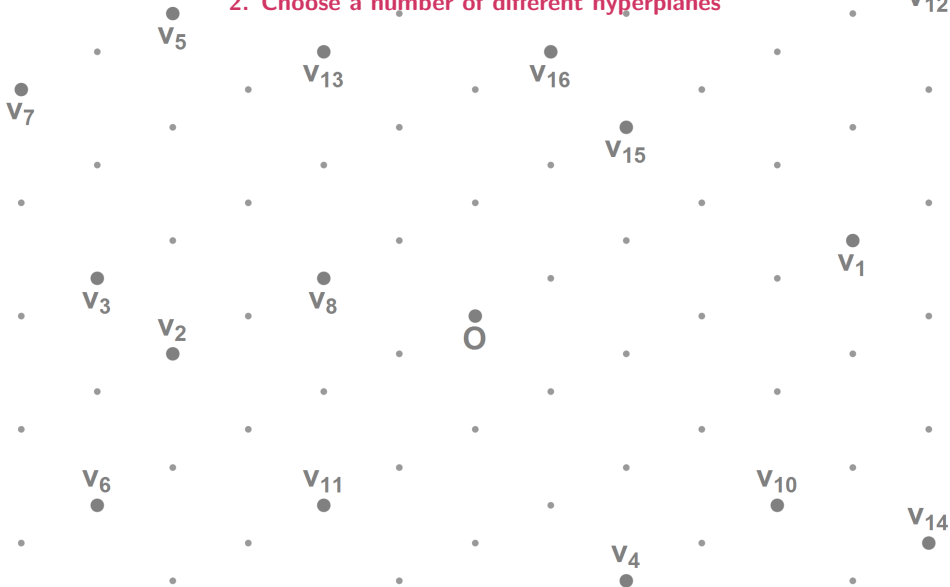
Sieving with LSH

1. Sample a list L of random lattice vectors



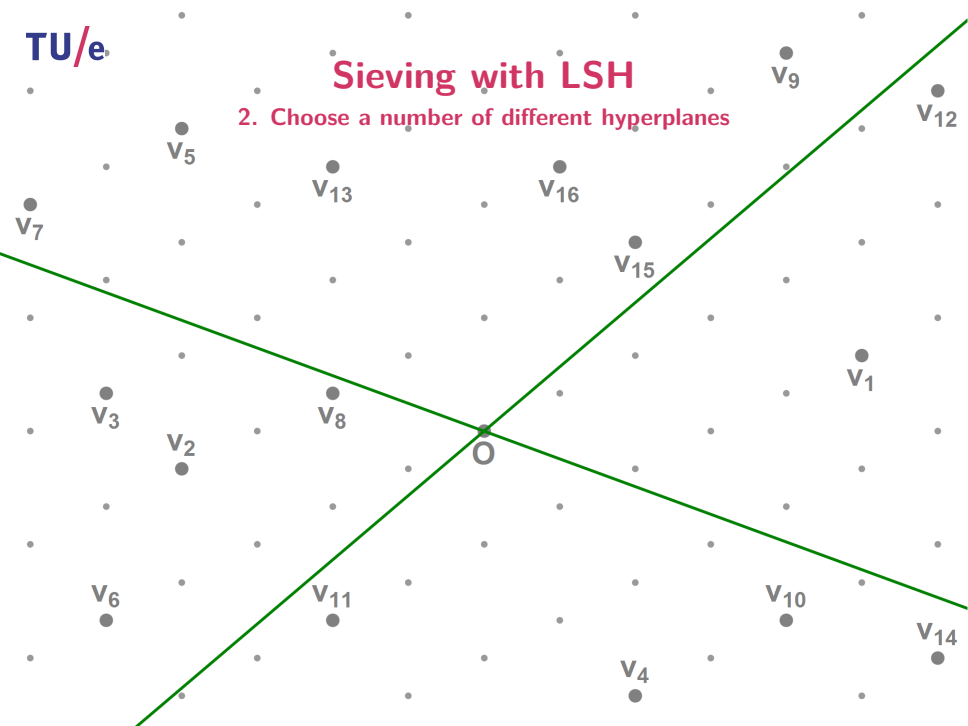
Sieving with LSH

2. Choose a number of different hyperplanes



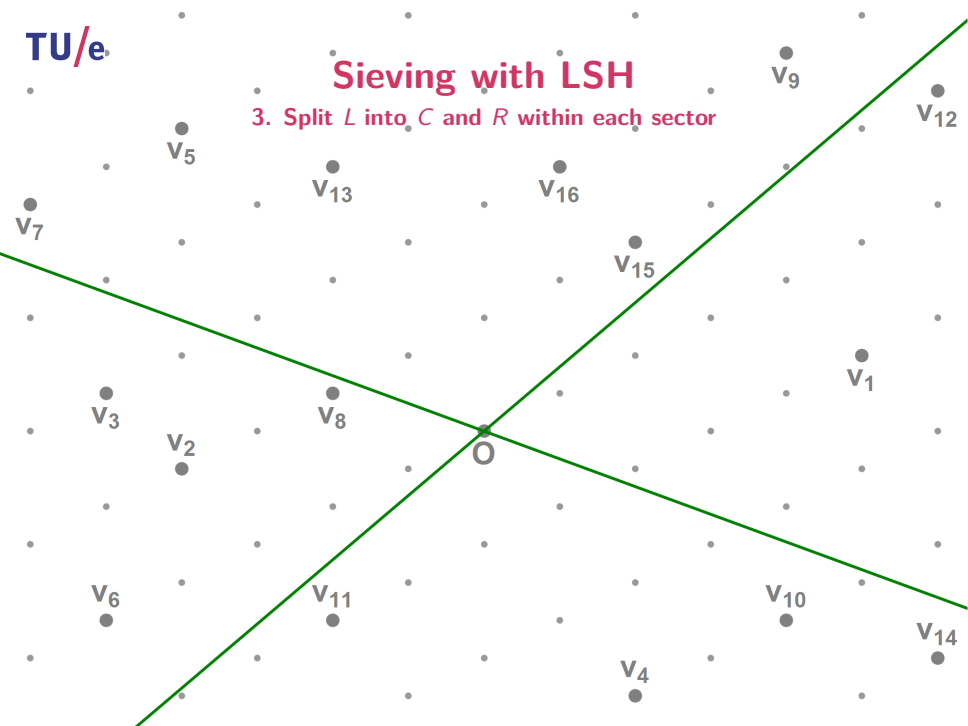
Sieving with LSH

2. Choose a number of different hyperplanes



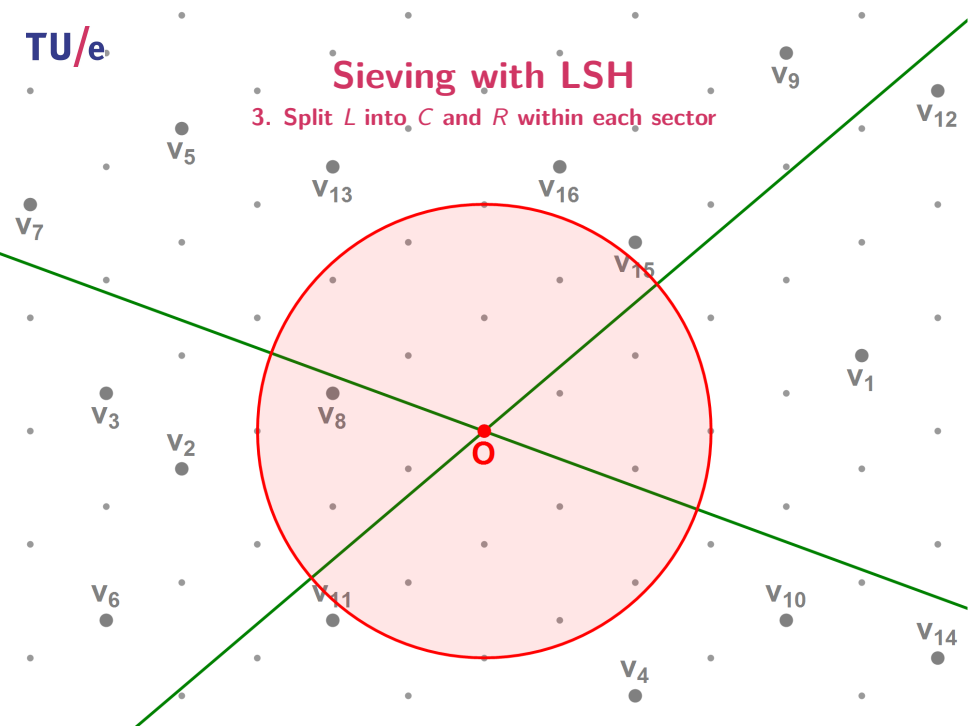
Sieving with LSH

3. Split L into C and R within each sector



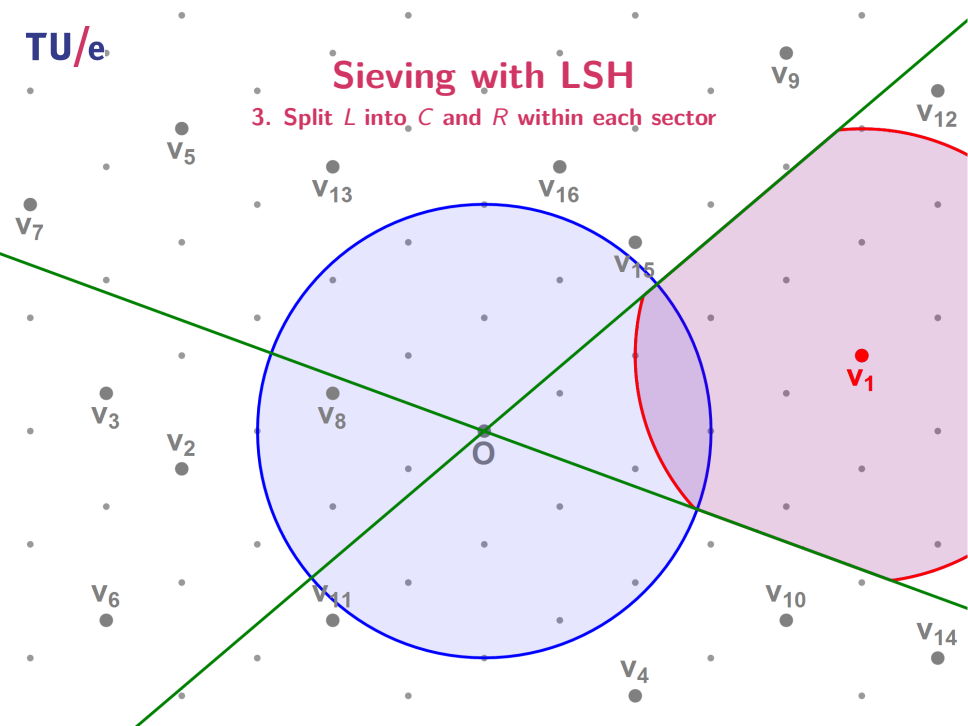
Sieving with LSH

3. Split L into C and R within each sector



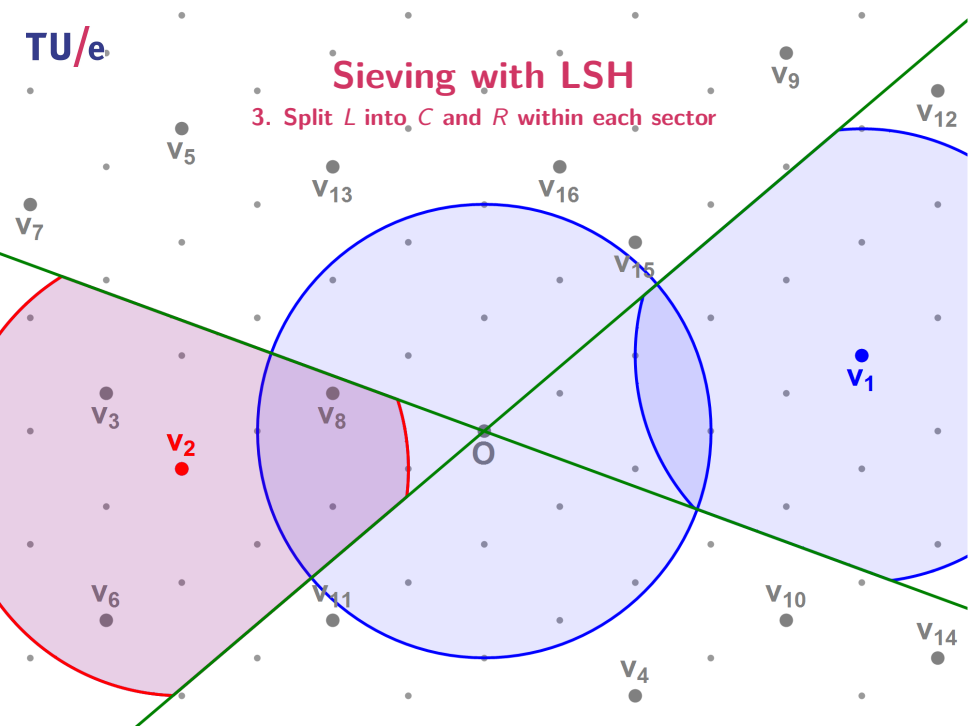
Sieving with LSH

3. Split L into C and R within each sector



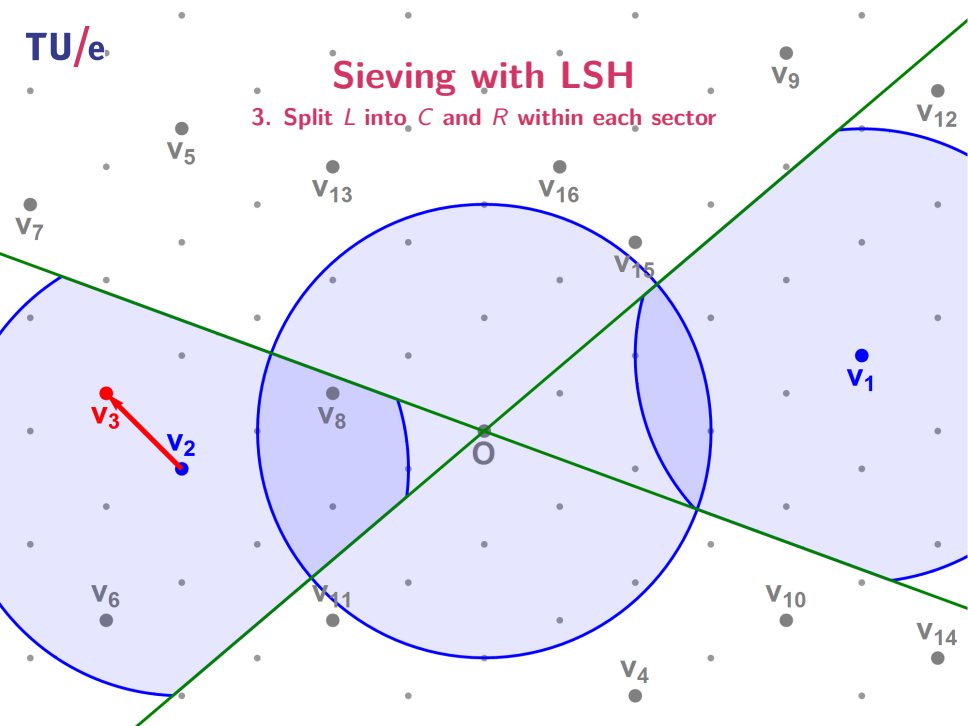
Sieving with LSH

3. Split L into C and R within each sector



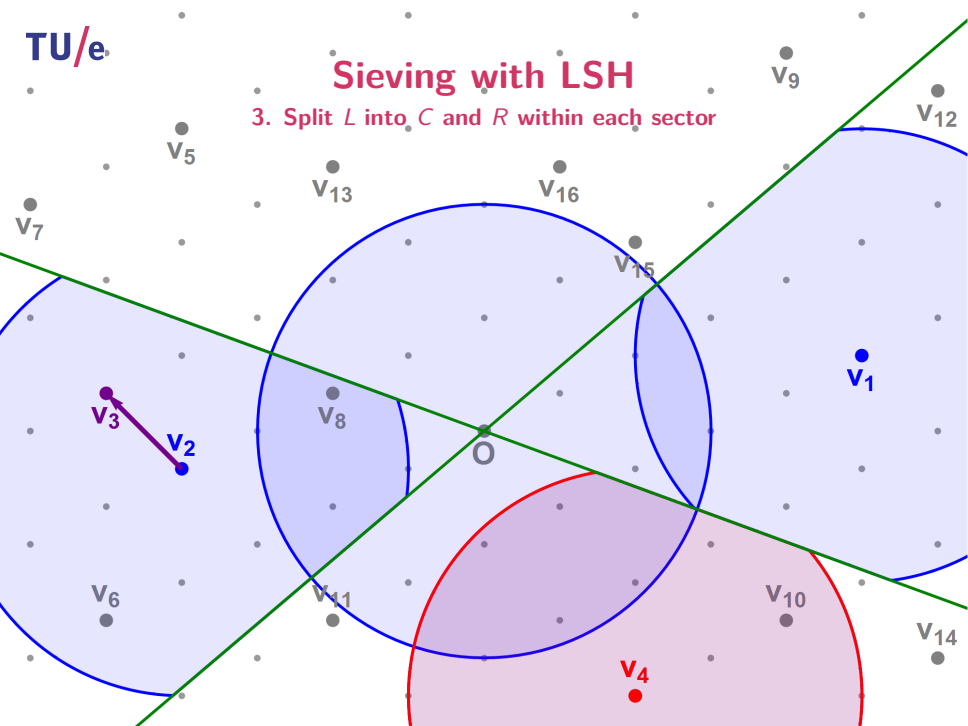
Sieving with LSH

3. Split L into C and R within each sector



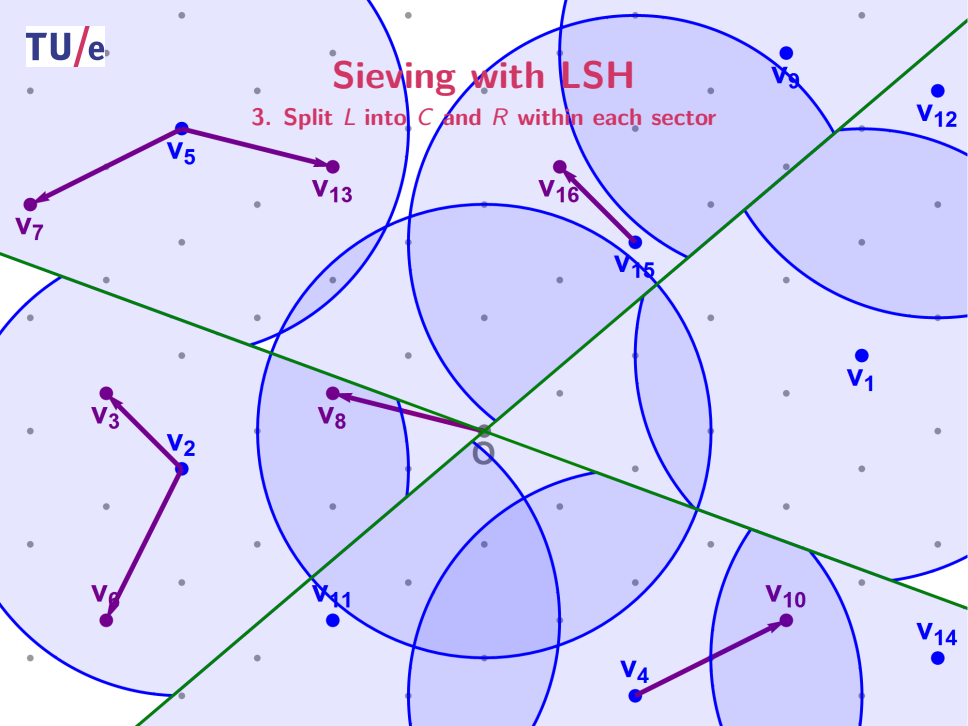
Sieving with LSH

3. Split L into C and R within each sector



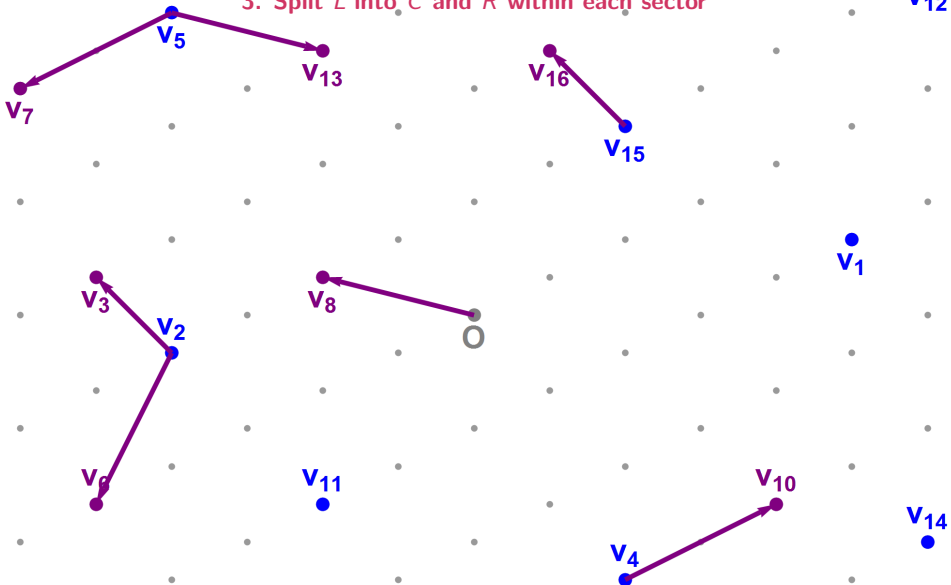
Sieving with LSH

3. Split L into C and R within each sector



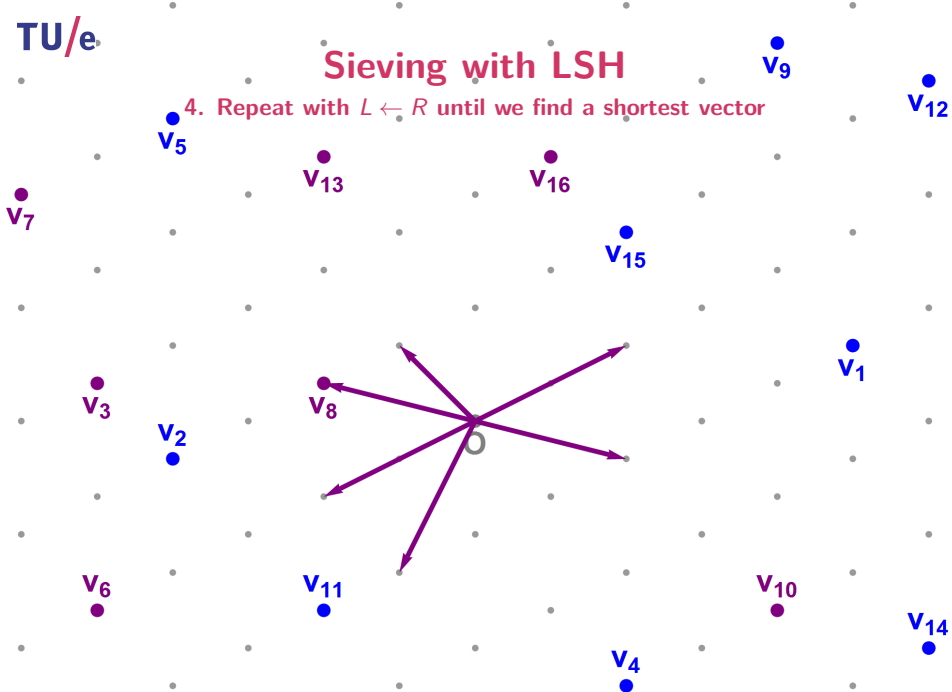
Sieving with LSH

3. Split L into C and R within each sector



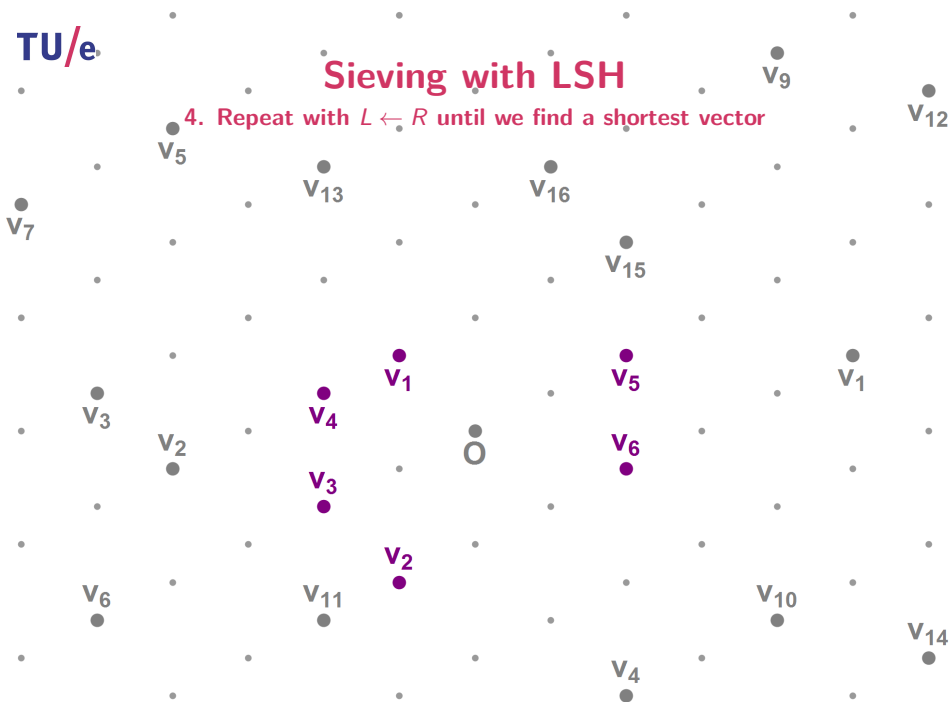
Sieving with LSH

4. Repeat with $L \leftarrow R$ until we find a shortest vector



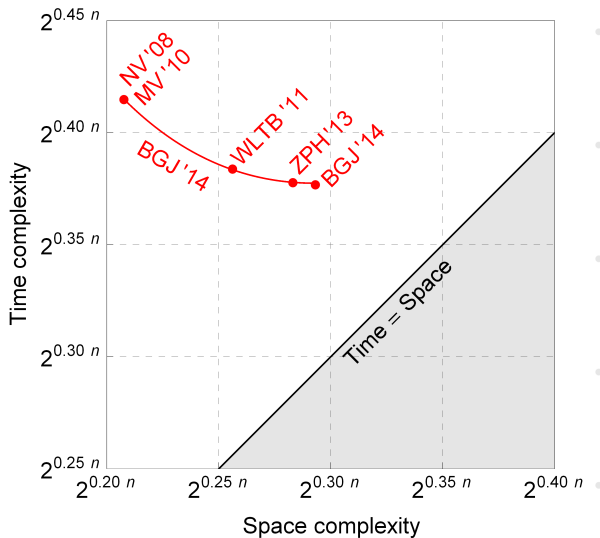
Sieving with LSH

4. Repeat with $L \leftarrow R$ until we find a shortest vector



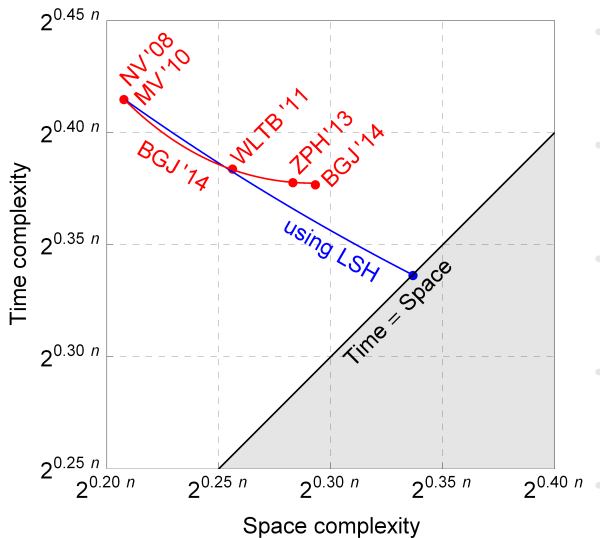
Sieving with LSH

Space/time trade-off



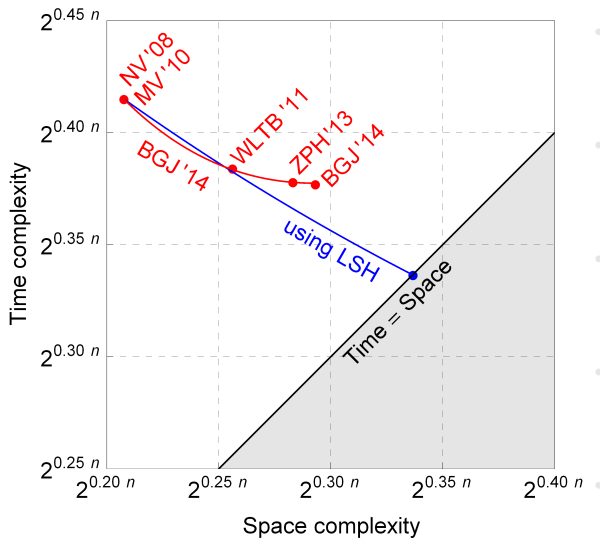
Sieving with LSH

Space/time trade-off



Sieving with LSH and Grover

Space/time trade-off



Sieving with LSH and Grover

Space/time trade-off

