CSE 548 / AMS 542: Analysis of Algorithms

Lecture 4
(Divide-and-Conquer Algorithms: Polynomial Multiplication)

Rezaul A. Chowdhury

Department of Computer Science
SUNY Stony Brook
Fall 2023

Coefficient Representation of Polynomials

$$A(x) = \sum_{j=0}^{n-1} a_j x^j$$

$$= a_0 + a_1 x + a_2 x^2 + \cdots + a_{n-1} x^{n-1}$$

$A(x)$ is a polynomial of degree bound n represented as a vector $a = (a_0, a_1, \cdots, a_{n-1})$ of coefficients.

The *degree* of $A(x)$ is k provided it is the largest integer such that a_k is nonzero. Clearly, $0 \le k \le n-1$.

Evaluating $A(x)$ at a given point:

Takes $\Theta(n)$ time using Horner's rule:

$$A(x_0) = a_0 + a_1 x_0 + a_2 (x_0)^2 + \cdots + a_{n-1} (x_0)^{n-1}$$

$$= a_0 + x_0 \Big(a_1 + x_0 \big(a_2 + \cdots + x_0 \big(a_{n-2} + x_0 (a_{n-1})\big) \cdots \big)\Big)$$

Coefficient Representation of Polynomials

Adding Two Polynomials:

Adding two polynomials of degree bound n takes $\Theta(n)$ time.

$$C(x) = A(x) + B(x)$$

where, $A(x) = \sum_{j=0}^{n-1} a_j x^j$ and $B(x) = \sum_{j=0}^{n-1} b_j x^j$.

Then $C(x) = \sum_{j=0}^{n-1} c_j x^j$, where, $c_j = a_j + b_j$ for $0 \le j \le n-1$.

Coefficient Representation of Polynomials

Multiplying Two Polynomials:

The product of two polynomials of degree bound n is another polynomial of degree bound $2n-1$.

$$C(x) = A(x)B(x)$$

where, $A(x) = \sum_{j=0}^{n-1} a_j x^j$ and $B(x) = \sum_{j=0}^{n-1} b_j x^j$.

Then $C(x) = \sum_{j=0}^{2n-2} c_j x^j$ where, $c_j = \sum_{k=0}^{j} a_k b_{j-k}$ for $0 \le j \le 2n-2$.

The coefficient vector $c = (c_0, c_1, \cdots, c_{2n-2})$, denoted by $c = a \otimes b$, is also called the *convolution* of vectors $a = (a_0, a_1, \cdots, a_{n-1})$ and $b = (b_0, b_1, \cdots, b_{n-1})$.

Clearly, straightforward evaluation of c takes $\Theta(n^2)$ time.

Polynomial Multiplication and Convolution

$$\boxed{a_0} + \boxed{a_1 x} + \boxed{a_2 x^2} + \boxed{a_3 x^3}$$

$$\boxed{b_3 x^3} + \boxed{b_2 x^2} + \boxed{b_1 x} + \boxed{b_0} \Rightarrow$$

$$\boxed{a_0 b_0}$$

$$(\quad a_0 b_0 \quad)$$

$$\begin{aligned}(a_0 + a_1 x + a_2 x^2 + a_3\, x^3) \times (b_0 + b_1 x + b_2 x^2 + b_3\, x^3) \\ = a_0 b_0 \\ + \cdots\end{aligned}$$

Polynomial Multiplication and Convolution

$$\boxed{a_0} + \boxed{a_1x} + \boxed{a_2x^2} + \boxed{a_3x^3}$$

$$\boxed{b_3x^3} + \boxed{b_2x^2} + \boxed{b_1x} + \boxed{b_0} \Rightarrow$$

$$\boxed{a_0b_1x} + \boxed{a_1b_0x}$$

$$(\quad a_0b_1 \quad + \quad a_1\,b_0 \quad)x$$

$$\begin{aligned}(a_0 + a_1x + a_2x^2 + a_3\,x^3) \times (b_0 + b_1x + b_2x^2 + b_3\,x^3)\\ &= a_0b_0\\ &+(a_0b_1 + a_1b_0)x\\ &+\cdots\end{aligned}$$

Polynomial Multiplication and Convolution

$$\boxed{a_0} + \boxed{a_1 x} + \boxed{a_2 x^2} + \boxed{a_3 x^3}$$

$$\boxed{b_3 x^3} + \boxed{b_2 x^2} + \boxed{b_1 x} + \boxed{b_0}$$

$$\boxed{a_0 b_2 x^2} + \boxed{a_1 b_1 x^2} + \boxed{a_2 b_0 x^2}$$

$$(\ a_0 b_2 \ + \ a_1 b_1 \ + \ a_2 b_0 \ )x^2$$

$$\begin{aligned}
(a_0 + a_1 x + a_2 x^2 + a_3 x^3) \times (b_0 + b_1 x + b_2 x^2 + b_3 x^3) \\
&= a_0 b_0 \\
&+ (a_0 b_1 + a_1 b_0)x \\
&+ (a_0 b_2 + a_1 b_1 + a_2 b_0)x^2 \\
&+ \cdots
\end{aligned}$$

Polynomial Multiplication and Convolution

$$
\begin{array}{ccccccc}
\boxed{a_0} & + & \boxed{a_1 x} & + & \boxed{a_2 x^2} & + & \boxed{a_3 x^3} \\
\boxed{b_3 x^3} & + & \boxed{b_2 x^2} & + & \boxed{b_1 x} & + & \boxed{b_0} \\
\boxed{a_0 b_3 x^3} & + & \boxed{a_1 b_2 x^3} & + & \boxed{a_2 b_1 x^3} & + & \boxed{a_3 b_0 x^3}
\end{array}
$$

$$(\quad a_0 b_3 \quad + \quad a_1 b_2 \quad + \quad a_2 b_1 \quad + \quad a_3 b_0 \quad) x^3$$

$$
\begin{aligned}
(a_0 + a_1 x + a_2 x^2 + a_3 x^3) \times (b_0 + b_1 x + b_2 x^2 + b_3 x^3) \\
&= a_0 b_0 \\
&+ (a_0 b_1 + a_1 b_0) x \\
&+ (a_0 b_2 + a_1 b_1 + a_2 b_0) x^2 \\
&+ (a_0 b_3 + a_1 b_2 + a_2 b_1 + a_3 b_0) x^3 \\
&+ \cdots
\end{aligned}
$$

Polynomial Multiplication and Convolution

$$\boxed{a_0} + \boxed{a_1 x} + \boxed{a_2 x^2} + \boxed{a_3 x^3}$$

$$\boxed{b_3 x^3} + \boxed{b_2 x^2} + \boxed{b_1 x} + \boxed{b_0}$$

$$\boxed{a_1 b_3 x^4} + \boxed{a_2 b_2 x^4} + \boxed{a_3 b_1 x^4}$$

$$(\quad a_1 b_3 \quad + \quad a_2 b_2 \quad + \quad a_3 b_1 \quad) x^4$$

$$\begin{aligned}
\left(a_0 + a_1 x + a_2 x^2 + a_3\, x^3\right) \times \left(b_0 + b_1 x + b_2 x^2 + b_3\, x^3\right) \\
&= a_0 b_0 \\
&+ (a_0 b_1 + a_1 b_0) x \\
&+ (a_0 b_2 + a_1 b_1 + a_2 b_0) x^2 \\
&+ (a_0 b_3 + a_1 b_2 + a_2 b_1 + a_3 b_0) x^3 \\
&+ (a_1 b_3 + a_2 b_2 + a_3 b_1) x^4 \\
&+ \cdots
\end{aligned}$$

Polynomial Multiplication and Convolution

$$\boxed{a_0} + \boxed{a_1 x} + \boxed{a_2 x^2} + \boxed{a_3 x^3}$$

$$\boxed{b_3 x^3} + \boxed{b_2 x^2} + \boxed{b_1 x} + \boxed{b_0}$$

$$\boxed{a_2 b_3 x^5} + \boxed{a_3 b_2 x^5}$$

$$(\quad a_2 b_3 \quad + \quad a_3 b_2 \quad)x^5$$

$$\begin{aligned}
\left(a_0 + a_1 x + a_2 x^2 + a_3\, x^3\right) \times \left(b_0 + b_1 x + b_2 x^2 + b_3\, x^3\right) \\
&= a_0 b_0 \\
&+ (a_0 b_1 + a_1 b_0) x \\
&+ (a_0 b_2 + a_1 b_1 + a_2 b_0) x^2 \\
&+ (a_0 b_3 + a_1 b_2 + a_2 b_1 + a_3 b_0) x^3 \\
&+ (a_1 b_3 + a_2 b_2 + a_3 b_1) x^4 \\
&+ (a_2 b_3 + a_3 b_2) x^5 \\
&+ \cdots
\end{aligned}$$

Polynomial Multiplication and Convolution

$$\boxed{a_0} + \boxed{a_1 x} + \boxed{a_2 x^2} + \boxed{a_3 x^3}$$

$$\boxed{b_3 x^3} + \boxed{b_2 x^2} + \boxed{b_1 x} + \boxed{b_0}$$

$$\boxed{a_3 b_3 x^6}$$

$$(\quad a_3 b_3 \quad) x^6$$

$$\begin{aligned}
\left(a_0 + a_1 x + a_2 x^2 + a_3\, x^3\right) \times \left(b_0 + b_1 x + b_2 x^2 + b_3\, x^3\right) \\
= a_0 b_0 \\
+(a_0 b_1 + a_1 b_0)x \\
+(a_0 b_2 + a_1 b_1 + a_2 b_0)x^2 \\
+(a_0 b_3 + a_1 b_2 + a_2 b_1 + a_3 b_0)x^3 \\
+(a_1 b_3 + a_2 b_2 + a_3 b_1)x^4 \\
+(a_2 b_3 + a_3 b_2)x^5 \\
+(a_3 b_3)x^6
\end{aligned}$$

Coefficient Representation of Polynomials

Multiplying Two Polynomials:

We can use Karatsuba's algorithm (assume n to be a power of 2):

$$A(x) = \sum_{j=0}^{n-1} a_j x^j = \sum_{j=0}^{\frac{n}{2}-1} a_j x^j + x^{\frac{n}{2}} \sum_{j=0}^{\frac{n}{2}-1} a_{\frac{n}{2}+j} x^j = A_1(x) + x^{\frac{n}{2}} A_2(x)$$

$$B(x) = \sum_{j=0}^{n-1} b_j x^j = \sum_{j=0}^{\frac{n}{2}-1} b_j x^j + x^{\frac{n}{2}} \sum_{j=0}^{\frac{n}{2}-1} b_{\frac{n}{2}+j} x^j = B_1(x) + x^{\frac{n}{2}} B_2(x)$$

Then $C(x) = A(x)B(x)$

$$= A_1(x)B_1(x) + x^{\frac{n}{2}}[A_1(x)B_2(x) + A_2(x)B_1(x)] + x^n\, A_2(x)B_2(x)$$

But $A_1(x)B_2(x) + A_2(x)B_1(x)$

$$= [A_1(x) + A_2(x)][B_1(x) + B_2(x)] - A_1(x)B_1(x) - A_2(x)B_2(x)$$

3 recursive multiplications of polynomials of degree bound $\frac{n}{2}$.

Similar recurrence as in Karatsuba's integer multiplication algorithm leading to a complexity of $\mathrm{O}\left(n^{\log_2 3}\right) = \mathrm{O}\left(n^{1.59}\right)$.

Point-Value Representation of Polynomials

A point-value representation of a polynomial $A(x)$ is a set of n point-value pairs $\{(x_0, y_0), (x_1, y_1), \dots, (x_{n-1}, y_{n-1})\}$ such that all x_k are distinct and $y_k = A(x_k)$ for $0 \le k \le n-1$.

A polynomial has many point-value representations.

Adding Two Polynomials:

Suppose we have point-value representations of two polynomials of degree bound n using the same set of n points.

$$A: \{(x_0, y_0^a), (x_1, y_1^a), \dots, (x_{n-1}, y_{n-1}^a)\}$$

$$B: \{(x_0, y_0^b), (x_1, y_1^b), \dots, (x_{n-1}, y_{n-1}^b)\}$$

If $C(x) = A(x) + B(x)$ then

$$C: \{(x_0, y_0^a + y_0^b), (x_1, y_1^a + y_1^b), \dots, (x_{n-1}, y_{n-1}^a + y_{n-1}^b)\}$$

Thus polynomial addition takes $\Theta(n)$ time.

Point-Value Representation of Polynomials

Multiplying Two Polynomials:

Suppose we have *extended* (why?) point-value representations of two polynomials of degree bound n using the same set of $2n$ points.

$$A: \{(x_0, y_0^a), (x_1, y_1^a), \dots, (x_{2n-1}, y_{2n-1}^a)\}$$

$$B: \{(x_0, y_0^b), (x_1, y_1^b), \dots, (x_{2n-1}, y_{2n-1}^b)\}$$

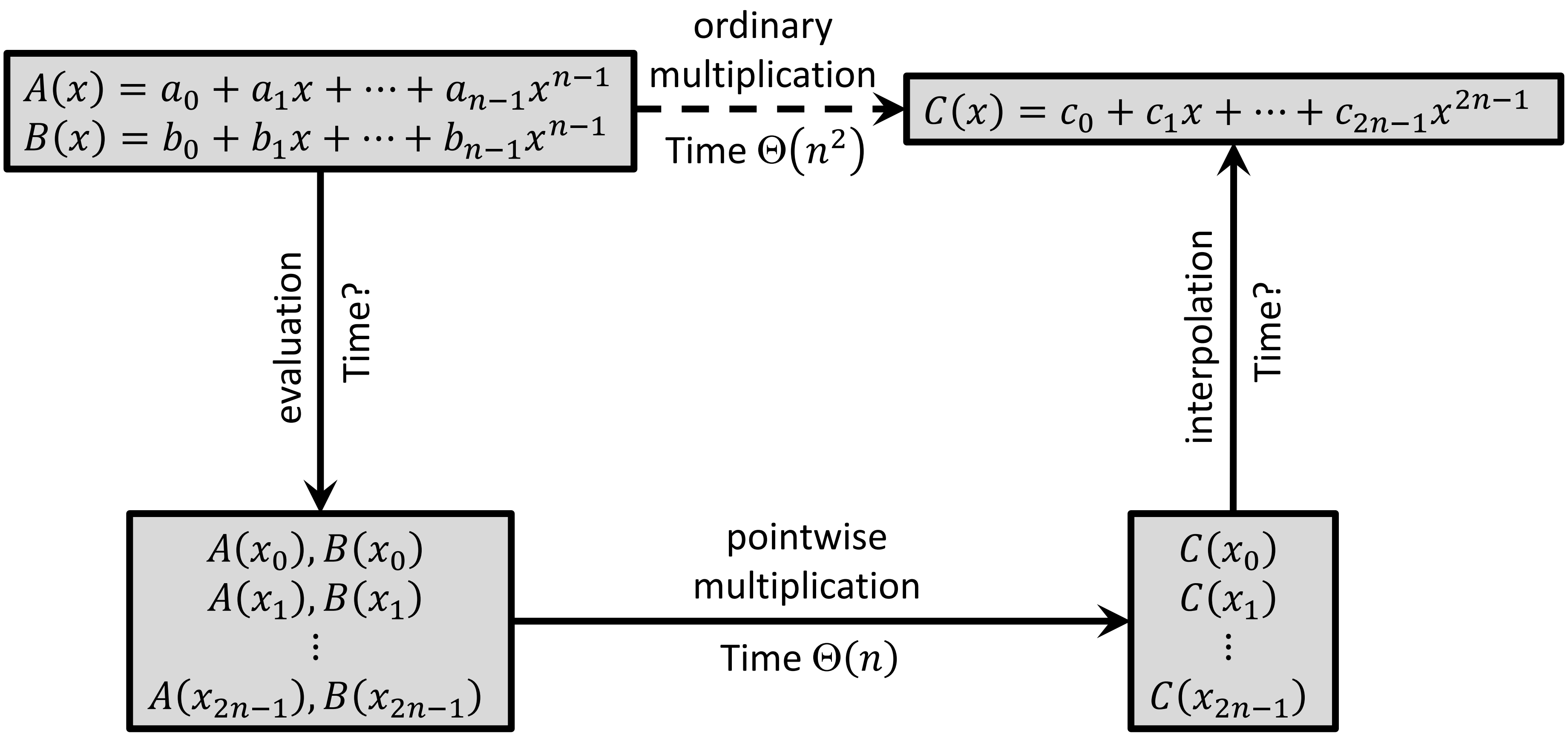
If $C(x) = A(x)B(x)$ then

$$C: \{(x_0, y_0^a y_0^b), (x_1, y_1^a y_1^b), \dots, (x_{2n-1}, y_{2n-1}^a y_{2n-1}^b)\}$$

Thus polynomial multiplication also takes only $\Theta(n)$ time!

(compare this with the $\Theta(n^2)$ time needed in the coefficient form)

Faster Polynomial Multiplication? (in Coefficient Form)



Faster Polynomial Multiplication? (in Coefficient Form)

Coefficient Representation $\Rightarrow$ Point-Value Representation:

We select any set of n distinct points $\{x_0, x_1, \ldots, x_{n-1}\}$, and evaluate $A(x_k)$ for $0 \le k \le n-1$.

Using Horner's rule this approach takes $\Theta(n^2)$ time.

Point-Value Representation $\Rightarrow$ Coefficient Representation:

We can interpolate using Lagrange's formula:

$$A(x) = \sum_{k=0}^{n-1} \frac{\prod_{j \ne k}(x - x_j)}{\prod_{j \ne k}(x_k - x_j)} y_k$$

This again takes $\Theta(n^2)$ time.

In both cases we need to do much better!

Coefficient Form $\Rightarrow$ Point-Value Form

A polynomial of degree bound n: $A(x) = a_0 + a_1 x + \cdots + a_{n-1} x^{n-1}$

A set of n distinct points: $\{x_0, x_1, \ldots, x_{n-1}\}$

Compute point-value form: $\{(x_0, A(x_0)), (x_1, A(x_1)), \ldots, (x_{n-1}, A(x_{n-1}))\}$

Using matrix notation:
$$\begin{bmatrix} A(x_0) \\ A(x_1) \\ \cdot \\ \cdot \\ \cdot \\ A(x_{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & x_0 & (x_0)^2 & \cdots & (x_0)^{n-1} \\ 1 & x_1 & (x_1)^2 & \cdots & (x_1)^{n-1} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ 1 & x_{n-1} & (x_{n-1})^2 & \cdots & (x_{n-1})^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ \cdot \\ \cdot \\ \cdot \\ a_{n-1} \end{bmatrix}$$

We want to choose the set of points in a way that simplifies the multiplication.

In the rest of the lecture on this topic we will assume:

n is a power of 2.

Given a Polynomial of Degree Bound 8 Find 8 Distinct Points to Efficiently Evaluate it at

$$\boxed{A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7}$$

	$A(x_0)$	$=$	a_0	$+$	$a_1 x_0$	$+$	$a_2(x_0)^2$	$+$	$a_3(x_0)^3$	$+$	$a_4(x_0)^4$	$+$	$a_5(x_0)^5$	$+$	$a_6(x_0)^6$	$+$	$a_7(x_0)^7$
	$A(x_1)$	$=$	a_0	$+$	$a_1 x_1$	$+$	$a_2(x_1)^2$	$+$	$a_3(x_1)^3$	$+$	$a_4(x_1)^4$	$+$	$a_5(x_1)^5$	$+$	$a_6(x_1)^6$	$+$	$a_7(x_1)^7$
	$A(x_2)$	$=$	a_0	$+$	$a_1 x_2$	$+$	$a_2(x_2)^2$	$+$	$a_3(x_2)^3$	$+$	$a_4(x_2)^4$	$+$	$a_5(x_2)^5$	$+$	$a_6(x_2)^6$	$+$	$a_7(x_2)^7$
	$A(x_3)$	$=$	a_0	$+$	$a_1 x_3$	$+$	$a_2(x_3)^2$	$+$	$a_3(x_3)^3$	$+$	$a_4(x_3)^4$	$+$	$a_5(x_3)^5$	$+$	$a_6(x_3)^6$	$+$	$a_7(x_3)^7$
$x_4 = -x_0$	$A(x_4)$	$=$	a_0	$+$	$a_1 x_4$	$+$	$a_2(x_4)^2$	$+$	$a_3(x_4)^3$	$+$	$a_4(x_4)^4$	$+$	$a_5(x_4)^5$	$+$	$a_6(x_4)^6$	$+$	$a_7(x_4)^7$
$x_5 = -x_1$	$A(x_5)$	$=$	a_0	$+$	$a_1 x_5$	$+$	$a_2(x_5)^2$	$+$	$a_3(x_5)^3$	$+$	$a_4(x_5)^4$	$+$	$a_5(x_5)^5$	$+$	$a_6(x_5)^6$	$+$	$a_7(x_5)^7$
$x_6 = -x_2$	$A(x_6)$	$=$	a_0	$+$	$a_1 x_6$	$+$	$a_2(x_6)^2$	$+$	$a_3(x_6)^3$	$+$	$a_4(x_6)^4$	$+$	$a_5(x_6)^5$	$+$	$a_6(x_6)^6$	$+$	$a_7(x_6)^7$
$x_7 = -x_3$	$A(x_7)$	$=$	a_0	$+$	$a_1 x_7$	$+$	$a_2(x_7)^2$	$+$	$a_3(x_7)^3$	$+$	$a_4(x_7)^4$	$+$	$a_5(x_7)^5$	$+$	$a_6(x_7)^6$	$+$	$a_7(x_7)^7$

STRATEGY: Set $x_{4+j} = -x_j$ for $0 \le j < 4$

Given a Polynomial of Degree Bound 8
Find 8 Distinct Points to Efficiently Evaluate it at

$$\boxed{A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7}$$

	$A(x_0)$	$=$	a_0	$+$	$a_1 x_0$	$+$	$a_2(x_0)^2$	$+$	$a_3(x_0)^3$	$+$	$a_4(x_0)^4$	$+$	$a_5(x_0)^5$	$+$	$a_6(x_0)^6$	$+$	$a_7(x_0)^7$
	$A(x_1)$	$=$	a_0	$+$	$a_1 x_1$	$+$	$a_2(x_1)^2$	$+$	$a_3(x_1)^3$	$+$	$a_4(x_1)^4$	$+$	$a_5(x_1)^5$	$+$	$a_6(x_1)^6$	$+$	$a_7(x_1)^7$
	$A(x_2)$	$=$	a_0	$+$	$a_1 x_2$	$+$	$a_2(x_2)^2$	$+$	$a_3(x_2)^3$	$+$	$a_4(x_2)^4$	$+$	$a_5(x_2)^5$	$+$	$a_6(x_2)^6$	$+$	$a_7(x_2)^7$
	$A(x_3)$	$=$	a_0	$+$	$a_1 x_3$	$+$	$a_2(x_3)^2$	$+$	$a_3(x_3)^3$	$+$	$a_4(x_3)^4$	$+$	$a_5(x_3)^5$	$+$	$a_6(x_3)^6$	$+$	$a_7(x_3)^7$
$x_4 = -x_0$	$A(-x_0)$	$=$	a_0	$-$	$a_1 x_0$	$+$	$a_2(x_0)^2$	$-$	$a_3(x_0)^3$	$+$	$a_4(x_0)^4$	$-$	$a_5(x_0)^5$	$+$	$a_6(x_0)^6$	$-$	$a_7(x_0)^7$
$x_5 = -x_1$	$A(-x_1)$	$=$	a_0	$-$	$a_1 x_1$	$+$	$a_2(x_1)^2$	$-$	$a_3(x_1)^3$	$+$	$a_4(x_1)^4$	$-$	$a_5(x_1)^5$	$+$	$a_6(x_1)^6$	$-$	$a_7(x_1)^7$
$x_6 = -x_2$	$A(-x_2)$	$=$	a_0	$-$	$a_1 x_2$	$+$	$a_2(x_2)^2$	$-$	$a_3(x_2)^3$	$+$	$a_4(x_2)^4$	$-$	$a_5(x_2)^5$	$+$	$a_6(x_2)^6$	$-$	$a_7(x_2)^7$
$x_7 = -x_3$	$A(-x_3)$	$=$	a_0	$-$	$a_1 x_3$	$+$	$a_2(x_3)^2$	$-$	$a_3(x_3)^3$	$+$	$a_4(x_3)^4$	$-$	$a_5(x_3)^5$	$+$	$a_6(x_3)^6$	$-$	$a_7(x_3)^7$

STRATEGY: Set $x_{4+j} = -x_j$ for $0 \leq j < 4$

Given a Polynomial of Degree Bound 8
Find 8 Distinct Points to Efficiently Evaluate it at

$$\boxed{A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7}$$

	$A(x_0)$	$=$	a_0	$+$	$a_2(x_0)^2$	$+$	$a_4(x_0)^4$	$+$	$a_6(x_0)^6$	$+$	$a_1 x_0$	$+$	$a_3(x_0)^3$	$+$	$a_5(x_0)^5$	$+$	$a_7(x_0)^7$
	$A(x_1)$	$=$	a_0	$+$	$a_2(x_1)^2$	$+$	$a_4(x_1)^4$	$+$	$a_6(x_1)^6$	$+$	$a_1 x_1$	$+$	$a_3(x_1)^3$	$+$	$a_5(x_1)^5$	$+$	$a_7(x_1)^7$
	$A(x_2)$	$=$	a_0	$+$	$a_2(x_2)^2$	$+$	$a_4(x_2)^4$	$+$	$a_6(x_2)^6$	$+$	$a_1 x_2$	$+$	$a_3(x_2)^3$	$+$	$a_5(x_2)^5$	$+$	$a_7(x_2)^7$
	$A(x_3)$	$=$	a_0	$+$	$a_2(x_3)^2$	$+$	$a_4(x_3)^4$	$+$	$a_6(x_3)^6$	$+$	$a_1 x_3$	$+$	$a_3(x_3)^3$	$+$	$a_5(x_3)^5$	$+$	$a_7(x_3)^7$
$x_4 = -x_0$	$A(-x_0)$	$=$	a_0	$+$	$a_2(x_0)^2$	$+$	$a_4(x_0)^4$	$+$	$a_6(x_0)^6$	$-$	$a_1 x_0$	$-$	$a_3(x_0)^3$	$-$	$a_5(x_0)^5$	$-$	$a_7(x_0)^7$
$x_5 = -x_1$	$A(-x_1)$	$=$	a_0	$+$	$a_2(x_1)^2$	$+$	$a_4(x_1)^4$	$+$	$a_6(x_1)^6$	$-$	$a_1 x_1$	$-$	$a_3(x_1)^3$	$-$	$a_5(x_1)^5$	$-$	$a_7(x_1)^7$
$x_6 = -x_2$	$A(-x_2)$	$=$	a_0	$+$	$a_2(x_2)^2$	$+$	$a_4(x_2)^4$	$+$	$a_6(x_2)^6$	$-$	$a_1 x_2$	$-$	$a_3(x_2)^3$	$-$	$a_5(x_2)^5$	$-$	$a_7(x_2)^7$
$x_7 = -x_3$	$A(-x_3)$	$=$	a_0	$+$	$a_2(x_3)^2$	$+$	$a_4(x_3)^4$	$+$	$a_6(x_3)^6$	$-$	$a_1 x_3$	$-$	$a_3(x_3)^3$	$-$	$a_5(x_3)^5$	$-$	$a_7(x_3)^7$

STRATEGY: Set $x_{4+j} = -x_j$ for $0 \leq j < 4$

Given a Polynomial of Degree Bound 8 Find 8 Distinct Points to Efficiently Evaluate it at

$$A(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7$$

	$A(x_0)$	$=$	$(a_0 + a_2(x_0)^2 + a_4(x_0)^4 + a_6(x_0)^6) + x_0(a_1 + a_3(x_0)^2 + a_5(x_0)^4 + a_7(x_0)^6)$
	$A(x_1)$	$=$	$(a_0 + a_2(x_1)^2 + a_4(x_1)^4 + a_6(x_1)^6) + x_1(a_1 + a_3(x_1)^2 + a_5(x_1)^4 + a_7(x_1)^6)$
	$A(x_2)$	$=$	$(a_0 + a_2(x_2)^2 + a_4(x_2)^4 + a_6(x_2)^6) + x_2(a_1 + a_3(x_2)^2 + a_5(x_2)^4 + a_7(x_2)^6)$
	$A(x_3)$	$=$	$(a_0 + a_2(x_3)^2 + a_4(x_3)^4 + a_6(x_3)^6) + x_3(a_1 + a_3(x_3)^2 + a_5(x_3)^4 + a_7(x_3)^6)$
$x_4 = -x_0$	$A(-x_0)$	$=$	$(a_0 + a_2(x_0)^2 + a_4(x_0)^4 + a_6(x_0)^6) - x_0(a_1 + a_3(x_0)^2 + a_5(x_0)^4 + a_7(x_0)^6)$
$x_5 = -x_1$	$A(-x_1)$	$=$	$(a_0 + a_2(x_1)^2 + a_4(x_1)^4 + a_6(x_1)^6) - x_1(a_1 + a_3(x_1)^2 + a_5(x_1)^4 + a_7(x_1)^6)$
$x_6 = -x_2$	$A(-x_2)$	$=$	$(a_0 + a_2(x_2)^2 + a_4(x_2)^4 + a_6(x_2)^6) - x_2(a_1 + a_3(x_2)^2 + a_5(x_2)^4 + a_7(x_2)^6)$
$x_7 = -x_3$	$A(-x_3)$	$=$	$(a_0 + a_2(x_3)^2 + a_4(x_3)^4 + a_6(x_3)^6) - x_3(a_1 + a_3(x_3)^2 + a_5(x_3)^4 + a_7(x_3)^6)$

STRATEGY: Set $x_{4+j} = -x_j$ for $0 \le j < 4$

Given a Polynomial of Degree Bound 8
Find 8 Distinct Points to Efficiently Evaluate it at

$$A(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7$$

$$A_{even}(x) = a_0 + a_2x + a_4x^2 + a_6x^3$$

$$A_{odd}(x) = a_1 + a_3x + a_5x^2 + a_7x^3$$

$$\begin{array}{llll}
 & A(x_0) & = & (a_0 + a_2(x_0^2) + a_4(x_0^2)^2 + a_6(x_0^2)^3) + x_0(a_1 + a_3(x_0^2) + a_5(x_0^2)^2 + a_7(x_0^2)^3) \\
 & A(x_1) & = & (a_0 + a_2(x_1^2) + a_4(x_1^2)^2 + a_6(x_1^2)^3) + x_1(a_1 + a_3(x_1^2) + a_5(x_1^2)^2 + a_7(x_1^2)^3) \\
 & A(x_2) & = & (a_0 + a_2(x_2^2) + a_4(x_2^2)^2 + a_6(x_2^2)^3) + x_2(a_1 + a_3(x_2^2) + a_5(x_2^2)^2 + a_7(x_2^2)^3) \\
 & A(x_3) & = & (a_0 + a_2(x_3^2) + a_4(x_3^2)^2 + a_6(x_3^2)^3) + x_3(a_1 + a_3(x_3^2) + a_5(x_3^2)^2 + a_7(x_3^2)^3) \\
\hline
x_4 = -x_0 & A(-x_0) & = & (a_0 + a_2(x_0^2) + a_4(x_0^2)^2 + a_6(x_0^2)^3) - x_0(a_1 + a_3(x_0^2) + a_5(x_0^2)^2 + a_7(x_0^2)^3) \\
x_5 = -x_1 & A(-x_1) & = & (a_0 + a_2(x_1^2) + a_4(x_1^2)^2 + a_6(x_1^2)^3) - x_1(a_1 + a_3(x_1^2) + a_5(x_1^2)^2 + a_7(x_1^2)^3) \\
x_6 = -x_2 & A(-x_2) & = & (a_0 + a_2(x_2^2) + a_4(x_2^2)^2 + a_6(x_2^2)^3) - x_2(a_1 + a_3(x_2^2) + a_5(x_2^2)^2 + a_7(x_2^2)^3) \\
x_7 = -x_3 & A(-x_3) & = & (a_0 + a_2(x_3^2) + a_4(x_3^2)^2 + a_6(x_3^2)^3) - x_3(a_1 + a_3(x_3^2) + a_5(x_3^2)^2 + a_7(x_3^2)^3)
\end{array}$$

STRATEGY: Set $x_{4+j} = -x_j$ for $0 \leq j < 4$

Given a Polynomial of Degree Bound 8
Find 8 Distinct Points to Efficiently Evaluate it at

$$A(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7$$

$$A_{even}(x) = a_0 + a_2x + a_4x^2 + a_6x^3$$

$$A_{odd}(x) = a_1 + a_3x + a_5x^2 + a_7x^3$$

	$A(x_0)$	$=$	$A_{even}(x_0^2)$	$+$	$x_0\, A_{odd}(x_0^2)$
	$A(x_1)$	$=$	$A_{even}(x_1^2)$	$+$	$x_1\, A_{odd}(x_1^2)$
	$A(x_2)$	$=$	$A_{even}(x_2^2)$	$+$	$x_2\, A_{odd}(x_2^2)$
	$A(x_3)$	$=$	$A_{even}(x_3^2)$	$+$	$x_3\, A_{odd}(x_3^2)$
$x_4 = -x_0$	$A(-x_0)$	$=$	$A_{even}(x_0^2)$	$-$	$x_0\, A_{odd}(x_0^2)$
$x_5 = -x_1$	$A(-x_1)$	$=$	$A_{even}(x_1^2)$	$-$	$x_1\, A_{odd}(x_1^2)$
$x_6 = -x_2$	$A(-x_2)$	$=$	$A_{even}(x_2^2)$	$-$	$x_2\, A_{odd}(x_2^2)$
$x_7 = -x_3$	$A(-x_3)$	$=$	$A_{even}(x_3^2)$	$-$	$x_3\, A_{odd}(x_3^2)$

STRATEGY: Set $x_{4+j} = -x_j$ for $0 \leq j < 4$

Given a Polynomial of Degree Bound 8
Find 8 Distinct Points to Efficiently Evaluate it at

$$\boxed{A(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7}$$

$$A_{even}(x) = a_0 + a_2x + a_4x^2 + a_6x^3$$

$$A_{odd}(x) = a_1 + a_3x + a_5x^2 + a_7x^3$$

	$A(x_0)$	$=$	$A_{even}(x_0^2)$	$+$	$x_0\, A_{odd}(x_0^2)$
	$A(x_1)$	$=$	$A_{even}(x_1^2)$	$+$	$x_1\, A_{odd}(x_1^2)$
	$A(x_2)$	$=$	$A_{even}(x_2^2)$	$+$	$x_2\, A_{odd}(x_2^2)$
	$A(x_3)$	$=$	$A_{even}(x_3^2)$	$+$	$x_3\, A_{odd}(x_3^2)$
$x_4 = -x_0$	$A(-x_0)$	$=$	$A_{even}(x_0^2)$	$-$	$x_0\, A_{odd}(x_0^2)$
$x_5 = -x_1$	$A(-x_1)$	$=$	$A_{even}(x_1^2)$	$-$	$x_1\, A_{odd}(x_1^2)$
$x_6 = -x_2$	$A(-x_2)$	$=$	$A_{even}(x_2^2)$	$-$	$x_2\, A_{odd}(x_2^2)$
$x_7 = -x_3$	$A(-x_3)$	$=$	$A_{even}(x_3^2)$	$-$	$x_3\, A_{odd}(x_3^2)$

STRATEGY: Set $x_{4+j} = -x_j$ for $0 \le j < 4$

We save roughly half the work.

Given a Polynomial of Degree Bound 2 Find 2 Distinct Points to Efficiently Evaluate it at

$$\boxed{A(x) = a_0 + a_1 x}$$

	$A(x_0)$	$=$	a_0	$+$	$a_1 x_0$
$x_1 = -x_0$	$A(x_1)$	$=$	a_0	$+$	$a_1 x_1$

STRATEGY: Set $x_{1+j} = -x_j$ for $0 \le j < 1$

Given a Polynomial of Degree Bound 2
Find 2 Distinct Points to Efficiently Evaluate it at

$$\boxed{A(x) = a_0 + a_1 x}$$

	$A(x_0)$	$=$	a_0	$+$	$a_1 x_0$
$x_1 = -x_0$	$A(-x_0)$	$=$	a_0	$-$	$a_1 x_0$

STRATEGY: Set $x_{1+j} = -x_j$ for $0 \leq j < 1$

Given a Polynomial of Degree Bound 2 Find 2 Distinct Points to Efficiently Evaluate it at

$$\boxed{A(x) = a_0 + a_1 x}$$

$x_0 = 1$	$A(x_0)$	$=$	a_0	$+$	a_1
$x_1 = -1$	$A(x_1)$	$=$	a_0	$-$	a_1

STRATEGY: We will evaluate any polynomial of degree bound 2 at

$$x_0 = 1$$
$$x_1 = -1$$

Given a Polynomial of Degree Bound 4 Find 4 Distinct Points to Efficiently Evaluate it at

$$A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

	$A(x_0)$	$=$	a_0	$+$	$a_1 x_0$	$+$	$a_2 (x_0)^2$	$+$	$a_3 (x_0)^3$
	$A(x_1)$	$=$	a_0	$+$	$a_1 x_1$	$+$	$a_2 (x_1)^2$	$+$	$a_3 (x_1)^3$
$x_2 = -x_0$	$A(x_2)$	$=$	a_0	$+$	$a_1 x_2$	$+$	$a_2 (x_2)^2$	$+$	$a_3 (x_2)^3$
$x_3 = -x_1$	$A(x_3)$	$=$	a_0	$+$	$a_1 x_3$	$+$	$a_2 (x_3)^2$	$+$	$a_3 (x_3)^3$

STRATEGY: Set $x_{2+j} = -x_j$ for $0 \leq j < 2$

Given a Polynomial of Degree Bound 4 Find 4 Distinct Points to Efficiently Evaluate it at

$$A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

	$A(x_0)$	$=$	a_0	$+$	$a_1 x_0$	$+$	$a_2 (x_0)^2$	$+$	$a_3 (x_0)^3$
	$A(x_1)$	$=$	a_0	$+$	$a_1 x_1$	$+$	$a_2 (x_1)^2$	$+$	$a_3 (x_1)^3$
$x_2 = -x_0$	$A(-x_0)$	$=$	a_0	$-$	$a_1 x_0$	$+$	$a_2 (x_0)^2$	$-$	$a_3 (x_0)^3$
$x_3 = -x_1$	$A(-x_1)$	$=$	a_0	$-$	$a_1 x_1$	$+$	$a_2 (x_1)^2$	$-$	$a_3 (x_1)^3$

STRATEGY: Set $x_{2+j} = -x_j$ for $0 \le j < 2$

Given a Polynomial of Degree Bound 4
Find 4 Distinct Points to Efficiently Evaluate it at

$$\boxed{A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3}$$

	$A(x_0)$	$=$	a_0	$+$	$a_2(x_0)^2$	$+$	$a_1 x_0$	$+$	$a_3(x_0)^3$
	$A(x_1)$	$=$	a_0	$+$	$a_2(x_1)^2$	$+$	$a_1 x_1$	$+$	$a_3(x_1)^3$
$x_2 = -x_0$	$A(-x_0)$	$=$	a_0	$+$	$a_2(x_0)^2$	$-$	$a_1 x_0$	$-$	$a_3(x_0)^3$
$x_3 = -x_1$	$A(-x_1)$	$=$	a_0	$+$	$a_2(x_1)^2$	$-$	$a_1 x_1$	$-$	$a_3(x_1)^3$

STRATEGY: Set $x_{2+j} = -x_j$ for $0 \le j < 2$

Given a Polynomial of Degree Bound 4
Find 4 Distinct Points to Efficiently Evaluate it at

$$A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

	$A(x_0)$	$=$	(	a_0	$+$	$a_2(x_0)^2$	)	$+$	x_0 (	a_1	$+$	$a_3(x_0)^2$	)
	$A(x_1)$	$=$	(	a_0	$+$	$a_2(x_1)^2$	)	$+$	x_1 (	a_1	$+$	$a_3(x_1)^2$	)
$x_2 = -x_0$	$A(-x_0)$	$=$	(	a_0	$+$	$a_2(x_0)^2$	)	$-$	x_0 (	a_1	$+$	$a_3(x_0)^2$	)
$x_3 = -x_1$	$A(-x_1)$	$=$	(	a_0	$+$	$a_2(x_1)^2$	)	$-$	x_1 (	a_1	$+$	$a_3(x_1)^2$	)

STRATEGY: Set $x_{2+j} = -x_j$ for $0 \leq j < 2$

Given a Polynomial of Degree Bound 4
Find 4 Distinct Points to Efficiently Evaluate it at

$$\boxed{A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3}$$

$$A_{even}(x) = a_0 + a_2 x$$

$$A_{odd}(x) = a_1 + a_3 x$$

	$A(x_0)$	$=$	$(a_0 + a_2(x_0^2)) + x_0(a_1 + a_3(x_0^2))$
	$A(x_1)$	$=$	$(a_0 + a_2(x_1^2)) + x_1(a_1 + a_3(x_1^2))$
$x_2 = -x_0$	$A(-x_0)$	$=$	$(a_0 + a_2(x_0^2)) - x_0(a_1 + a_3(x_0^2))$
$x_3 = -x_1$	$A(-x_1)$	$=$	$(a_0 + a_2(x_1^2)) - x_1(a_1 + a_3(x_1^2))$

STRATEGY: Set $x_{2+j} = -x_j$ for $0 \leq j < 2$

Given a Polynomial of Degree Bound 4 Find 4 Distinct Points to Efficiently Evaluate it at

$$A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3$$

$$A_{even}(x) = a_0 + a_2 x$$

$$A_{odd}(x) = a_1 + a_3 x$$

	$A(x_0)$	$=$	$A_{even}(x_0^2)$	$+$	$x_0\, A_{odd}(x_0^2)$
	$A(x_1)$	$=$	$A_{even}(x_1^2)$	$+$	$x_1\, A_{odd}(x_1^2)$
$x_2 = -x_0$	$A(-x_0)$	$=$	$A_{even}(x_0^2)$	$-$	$x_0\, A_{odd}(x_0^2)$
$x_3 = -x_1$	$A(-x_1)$	$=$	$A_{even}(x_1^2)$	$-$	$x_1\, A_{odd}(x_1^2)$

STRATEGY: Set $x_{2+j} = -x_j$ for $0 \leq j < 2$

Given a Polynomial of Degree Bound 4
Find 4 Distinct Points to Efficiently Evaluate it at

$$\boxed{A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3}$$

$$A_{even}(x) = a_0 + a_2 x$$

$$A_{odd}(x) = a_1 + a_3 x$$

	$A(x_0)$	$=$	$A_{even}(x_0^2)$	$+$	$x_0\, A_{odd}(x_0^2)$
	$A(x_1)$	$=$	$A_{even}(x_1^2)$	$+$	$x_1\, A_{odd}(x_1^2)$
$x_2 = -x_0$	$A(-x_0)$	$=$	$A_{even}(x_0^2)$	$-$	$x_0\, A_{odd}(x_0^2)$
$x_3 = -x_1$	$A(-x_1)$	$=$	$A_{even}(x_1^2)$	$-$	$x_1\, A_{odd}(x_1^2)$

STRATEGY: Set $x_{2+j} = -x_j$ for $0 \leq j < 2$

Given a Polynomial of Degree Bound 4
Find 4 Distinct Points to Efficiently Evaluate it at

$$\boxed{A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3}$$

$$A_{even}(x) = a_0 + a_2 x$$

$$A_{odd}(x) = a_1 + a_3 x$$

$$\begin{array}{lccccc}
 & A(x_0) & = & A_{even}(x_0^2) & + & x_0\, A_{odd}(x_0^2) \\
 & A(x_1) & = & A_{even}(x_1^2) & + & x_1\, A_{odd}(x_1^2) \\
\hline
x_2 = -x_0 & A(-x_0) & = & A_{even}(x_0^2) & - & x_0\, A_{odd}(x_0^2) \\
x_3 = -x_1 & A(-x_1) & = & A_{even}(x_1^2) & - & x_1\, A_{odd}(x_1^2)
\end{array}$$

Observe that we evaluate both $A_{even}(x)$ and $A_{odd}(x)$ at $x = x_0^2$ and $x = x_1^2$.

But we decided to always evaluate polynomials of degree bound 2 at $x = 1$ and $x = -1$.

So, $x_0^2 = 1 \Rightarrow x_0 = 1$ and $x_1^2 = -1 \Rightarrow x_1 = \sqrt{-1} = i$.

Given a Polynomial of Degree Bound 4 Find 4 Distinct Points to Efficiently Evaluate it at

$$\boxed{A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3}$$

$$A_{even}(x) = a_0 + a_2 x$$

$$A_{odd}(x) = a_1 + a_3 x$$

	$A(x_0)$	$=$	$A_{even}(x_0^2)$	$+$	$x_0\, A_{odd}(x_0^2)$
	$A(x_1)$	$=$	$A_{even}(x_1^2)$	$+$	$x_1\, A_{odd}(x_1^2)$
$x_2 = -x_0$	$A(-x_0)$	$=$	$A_{even}(x_0^2)$	$-$	$x_0\, A_{odd}(x_0^2)$
$x_3 = -x_1$	$A(-x_1)$	$=$	$A_{even}(x_1^2)$	$-$	$x_1\, A_{odd}(x_1^2)$

So, we evaluate any polynomial of degree bound 4 at

$x_0 = 1,\ x_1 = i$

and

$x_2 = -x_0 = -1,\ x_3 = -x_1 = -i$

Given a Polynomial of Degree Bound 8
Find 8 Distinct Points to Efficiently Evaluate it at

$$A(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7$$

$$A_{even}(x) = a_0 + a_2x + a_4x^2 + a_6x^3$$
$$A_{odd}(x) = a_1 + a_3x + a_5x^2 + a_7x^3$$

	$A(x_0)$	$=$	$A_{even}(x_0^2)$	$+$	$x_0\ A_{odd}(x_0^2)$
	$A(x_1)$	$=$	$A_{even}(x_1^2)$	$+$	$x_1\ A_{odd}(x_1^2)$
	$A(x_2)$	$=$	$A_{even}(x_2^2)$	$+$	$x_2\ A_{odd}(x_2^2)$
	$A(x_3)$	$=$	$A_{even}(x_3^2)$	$+$	$x_3\ A_{odd}(x_3^2)$
$x_4 = -x_0$	$A(-x_0)$	$=$	$A_{even}(x_0^2)$	$-$	$x_0\ A_{odd}(x_0^2)$
$x_5 = -x_1$	$A(-x_1)$	$=$	$A_{even}(x_1^2)$	$-$	$x_1\ A_{odd}(x_1^2)$
$x_6 = -x_2$	$A(-x_2)$	$=$	$A_{even}(x_2^2)$	$-$	$x_2\ A_{odd}(x_2^2)$
$x_7 = -x_3$	$A(-x_3)$	$=$	$A_{even}(x_3^2)$	$-$	$x_3\ A_{odd}(x_3^2)$

Observe that we evaluate both $A_{even}(x)$ and $A_{odd}(x)$ at $x = x_0^2$, $x = x_1^2$, $x = x_2^2$ and $x = x_3^2$.

But we decided to always evaluate polynomials of degree bound 4 at $x = 1$, $x = i$, $x = -1$ and $x = -i$.

So, $x_0^2 = 1 \Rightarrow x_0 = 1$, $x_1^2 = i \Rightarrow x_1 = \frac{1+i}{\sqrt{2}}$, $x_2^2 = -1 \Rightarrow x_2 = i$, and $x_3^2 = -i \Rightarrow x_3 = \frac{-1+i}{\sqrt{2}}$.

Given a Polynomial of Degree Bound 8
Find 8 Distinct Points to Efficiently Evaluate it at

$$A(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4 + a_5x^5 + a_6x^6 + a_7x^7$$

$$A_{even}(x) = a_0 + a_2x + a_4x^2 + a_6x^3$$
$$A_{odd}(x) = a_1 + a_3x + a_5x^2 + a_7x^3$$

	$A(x_0)$	$=$	$A_{even}(x_0^2)$	$+$	$x_0\, A_{odd}(x_0^2)$
	$A(x_1)$	$=$	$A_{even}(x_1^2)$	$+$	$x_1\, A_{odd}(x_1^2)$
	$A(x_2)$	$=$	$A_{even}(x_2^2)$	$+$	$x_2\, A_{odd}(x_2^2)$
	$A(x_3)$	$=$	$A_{even}(x_3^2)$	$+$	$x_3\, A_{odd}(x_3^2)$
$x_4 = -x_0$	$A(-x_0)$	$=$	$A_{even}(x_0^2)$	$-$	$x_0\, A_{odd}(x_0^2)$
$x_5 = -x_1$	$A(-x_1)$	$=$	$A_{even}(x_1^2)$	$-$	$x_1\, A_{odd}(x_1^2)$
$x_6 = -x_2$	$A(-x_2)$	$=$	$A_{even}(x_2^2)$	$-$	$x_2\, A_{odd}(x_2^2)$
$x_7 = -x_3$	$A(-x_3)$	$=$	$A_{even}(x_3^2)$	$-$	$x_3\, A_{odd}(x_3^2)$

So, we evaluate any polynomial of degree bound 8 at

$$x_0 = 1,\ x_1 = \frac{1+i}{\sqrt{2}},\ x_2 = i,\ x_3 = \frac{-1+i}{\sqrt{2}}$$

and

$$x_4 = -x_0 = -1,\ x_5 = -x_1 = -\frac{1+i}{\sqrt{2}},\ x_6 = -x_2 = -i,\ x_7 = -x_3 = -\frac{-1+i}{\sqrt{2}}$$

Given a Polynomial of Degree Bound $n = 2^k$ Find $n = 2^k$ Distinct Points to Efficiently Evaluate it at

degree bound	how did we find the points to evaluate the polynomial at?	the points	point property
2^1	… … …	$1,$ -1	all 2nd roots of unity
2^2	take positive and negative square roots of points used for degree bound 2^1 which are already the 2nd roots of unity	$1, \quad i,$ $-1, \quad -i$	all 4th roots of unity
2^3	take positive and negative square roots of points used for degree bound 2^2 which are already the 4th roots of unity	$1, \quad \frac{1+i}{\sqrt{2}}, \quad i, \quad \frac{-1+i}{\sqrt{2}},$ $-1, \quad -\frac{1+i}{\sqrt{2}}, \quad -i, \quad -\frac{-1+i}{\sqrt{2}}$	all 8th roots of unity
2^4	take positive and negative square roots of points used for degree bound 2^3 which are already the 8th roots of unity	$1, \quad \frac{\sqrt{2+\sqrt{2}}}{2} + i\frac{\sqrt{2-\sqrt{2}}}{2}, \quad \ldots, \quad \ldots,$ $-1, \quad -\left(\frac{\sqrt{2+\sqrt{2}}}{2} + i\frac{\sqrt{2-\sqrt{2}}}{2}\right), \quad \ldots, \quad \ldots$	all 16th roots of unity
… … …	… … …	… … …	… … …
2^{k-1}	take positive and negative square roots of points used for degree bound 2^{k-2} which are already the 2^{k-2}th roots of unity	… … …	all 2^{k-1}th roots of unity
$n = 2^k$	take positive and negative square roots of points used for degree bound 2^{k-1} which are already the 2^{k-1}th roots of unity	… … …	all 2^kth roots of unity (i.e., n^{th} roots of unity)

How to Find all n^{th} Roots of Unity

The n^{th} roots of unity are: $1,\ \omega_n, (\omega_n)^2, (\omega_n)^3, \dots \dots \dots, (\omega_n)^{n-1}$,
where $\omega_n = \cos\frac{2\Pi}{n} + i\sin\frac{2\Pi}{n} = e^{\frac{2\Pi i}{n}}$ is known as the primitive n^{th} roots of unity.

The result above can be derived using Euler's Formula.

Euler's Formula: For any real number α, $\cos\alpha + i\sin\alpha = e^{i\alpha}$

Euler's formula follows very easily from the following three power series each of which holds for $-\infty < \alpha < +\infty$:

$$\cos\alpha = 1 - \frac{\alpha^2}{2!} + \frac{\alpha^4}{4!} - \frac{\alpha^6}{6!} + \frac{\alpha^8}{8!} - \cdots$$

$$\sin\alpha = \alpha - \frac{\alpha^3}{3!} + \frac{\alpha^5}{5!} - \frac{\alpha^7}{7!} + \frac{\alpha^9}{9!} - \cdots$$

$$e^\alpha = 1 + \alpha + \frac{\alpha^2}{2!} + \frac{\alpha^3}{3!} + \frac{\alpha^4}{4!} + \frac{\alpha^5}{5!} + \frac{\alpha^6}{6!} + \frac{\alpha^7}{7!} + \frac{\alpha^8}{8!} + \cdots$$

How to Find all n^{th} Roots of Unity

Observe that for (any) real numbers α and p,

$$(\cos\alpha + i\sin\alpha)^p = \left(e^{i\alpha}\right)^p = e^{i(p\alpha)} = \cos(p\alpha) + i\sin(p\alpha)$$

Also observe that for any integer k, $\cos(k \times 2\Pi) + i\sin(k \times 2\Pi) = 1 + i \times 0 = 1$

Then the n^{th} root of 1 (unity) is:

$$1^{\frac{1}{n}} = (\cos(k \times 2\Pi) + i\sin(k \times 2\Pi))^{\frac{1}{n}} = \cos\left(k \times \frac{2\Pi}{\mathrm{n}}\right) + i\sin\left(k \times \frac{2\Pi}{n}\right)$$

Observe that $\cos\left(k \times \frac{2\Pi}{\mathrm{n}}\right) + i\sin\left(k \times \frac{2\Pi}{n}\right)$ takes n distinct values for $0 \leq k < n$, and then simply repeats those values for $k < 0$ and $k \geq n$.

When $k = 1$, we have: $\cos\left(k \times \frac{2\Pi}{\mathrm{n}}\right) + i\sin\left(k \times \frac{2\Pi}{n}\right) = \cos\left(\frac{2\Pi}{\mathrm{n}}\right) + i\sin\left(\frac{2\Pi}{n}\right)$
$= \omega_n =$ primitive n^{th} root of 1.

Clearly, for any k, $\cos\left(k \times \frac{2\Pi}{n}\right) + i\sin\left(k \times \frac{2\Pi}{n}\right) = \left(\cos\left(\frac{2\Pi}{\mathrm{n}}\right) + i\sin\left(\frac{2\Pi}{n}\right)\right)^k = (\omega_n)^k$

Hence, $1^{\frac{1}{n}} = \cos\left(k \times \frac{2\Pi}{\mathrm{n}}\right) + i\sin\left(k \times \frac{2\Pi}{n}\right) = (\omega_n)^k$, for $k = 0, 1, 2, \ldots, n-1$.

In other words, the n^{th} roots of 1 (unity) are: $1,\ \omega_n, (\omega_n)^2, (\omega_n)^3, \ldots \ldots \ldots, (\omega_n)^{n-1}$

Coefficient Form $\Rightarrow$ Point-Value Form

For a polynomial of degree bound $n = 2^k$, we need to apply the trick recursively at most $\log n = k$ times.

We choose $x_0 = 1 = \omega_n^0$ and set $x_j = \omega_n^j$ for $1 \le j \le n-1$.

Then we compute the following product:

$$\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_{n-1} \end{bmatrix} = \begin{bmatrix} A(1) \\ A(\omega_n) \\ A(\omega_n^2) \\ \cdot \\ \cdot \\ A(\omega_n^{n-1}) \end{bmatrix} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & (\omega_n)^2 & \cdots & (\omega_n)^{n-1} \\ 1 & \omega_n^2 & (\omega_n^2)^2 & \cdots & (\omega_n^2)^{n-1} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ 1 & \omega_n^{n-1} & (\omega_n^{n-1})^2 & \cdots & (\omega_n^{n-1})^{n-1} \end{bmatrix} \begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_{n-1} \end{bmatrix}$$

The vector $y = (y_0, y_1, \cdots, y_{n-1})$ is called the *discrete Fourier transform* (DFT) of $(a_0, a_1, \cdots, a_{n-1})$.

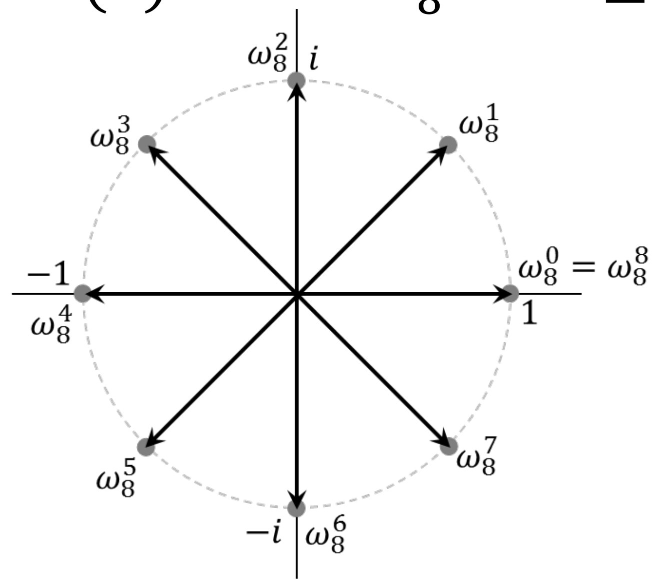
This method of computing DFT is called the *fast Fourier transform* (FFT) method.

Coefficient Form $\Rightarrow$ Point-Value Form

Example: For $n = 2^3 = 8$:

$A(x) = a_0 + a_1 x + a_2 x^2 + a_3 x^3 + a_4 x^4 + a_5 x^5 + a_6 x^6 + a_7 x^7$

We need to evaluate $A(x)$ at $x = \omega_8^i$ for $0 \le i < 8$.



complex 8^{th} roots of unity

Now $A(x) = A_{even}\left(x^2\right) + x \cdot A_{odd}\left(x^2\right)$,

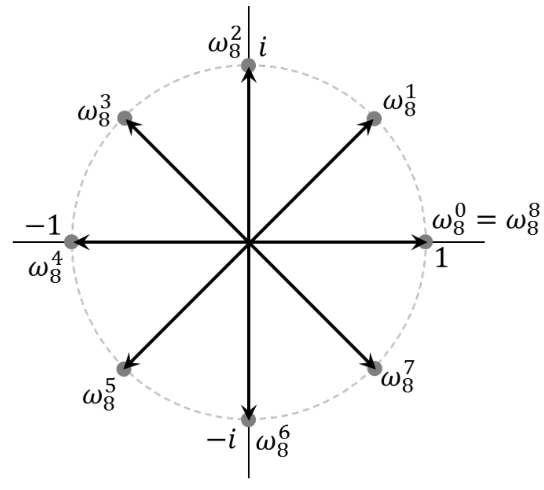
where $A_{even}(y) = a_0 + a_2 y + a_4 y^2 + a_6 y^3$

and $A_{odd}(y) = a_1 + a_3 y + a_5 y^2 + a_7 y^3$

Coefficient Form ⇒ Point-Value Form

Observe that:

$$\omega_8^0 = \omega_8^8 \quad = \omega_4^0$$
$$\omega_8^2 = \omega_8^{10} \quad = \omega_4^1$$
$$\omega_8^4 = \omega_8^{12} \quad = \omega_4^2$$
$$\omega_8^6 = \omega_8^{14} \quad = \omega_4^3$$



Also:

$$\omega_8^4 = -\omega_8^0$$
$$\omega_8^5 = -\omega_8^1$$
$$\omega_8^6 = -\omega_8^2$$
$$\omega_8^7 = -\omega_8^3$$

$$A(\omega_8^0) \; = A_{even}(\omega_8^0) + \omega_8^0 \cdot A_{odd}(\omega_8^0) \quad = A_{even}(\omega_4^0) + \omega_8^0 \cdot A_{odd}(\omega_4^0)$$
$$A(\omega_8^1) \; = A_{even}(\omega_8^2) + \omega_8^1 \cdot A_{odd}(\omega_8^2) \quad = A_{even}(\omega_4^1) + \omega_8^1 \cdot A_{odd}(\omega_4^1)$$
$$A(\omega_8^2) \; = A_{even}(\omega_8^4) + \omega_8^2 \cdot A_{odd}(\omega_8^4) \quad = A_{even}(\omega_4^2) + \omega_8^2 \cdot A_{odd}(\omega_4^2)$$
$$A(\omega_8^3) \; = A_{even}(\omega_8^6) + \omega_8^3 \cdot A_{odd}(\omega_8^6) \quad = A_{even}(\omega_4^3) + \omega_8^3 \cdot A_{odd}(\omega_4^3)$$

$$A(\omega_8^4) \; = A_{even}(\omega_8^8) + \omega_8^4 \cdot A_{odd}(\omega_8^8) \quad = A_{even}(\omega_4^0) - \omega_8^0 \cdot A_{odd}(\omega_4^0)$$
$$A(\omega_8^5) \; = A_{even}(\omega_8^{10}) + \omega_8^5 \cdot A_{odd}(\omega_8^{10}) \; = A_{even}(\omega_4^1) - \omega_8^1 \cdot A_{odd}(\omega_4^1)$$
$$A(\omega_8^6) \; = A_{even}(\omega_8^{12}) + \omega_8^6 \cdot A_{odd}(\omega_8^{12}) \; = A_{even}(\omega_4^2) - \omega_8^2 \cdot A_{odd}(\omega_4^2)$$
$$A(\omega_8^7) \; = A_{even}(\omega_8^{14}) + \omega_8^7 \cdot A_{odd}(\omega_8^{14}) \; = A_{even}(\omega_4^3) - \omega_8^3 \cdot A_{odd}(\omega_4^3)$$

Coefficient Form $\Rightarrow$ Point-Value Form

Rec-FFT ($(a_0, a_1, \ldots, a_{n-1})$) $\quad$ { $n = 2^k$ *for integer* $k \geq 0$ }

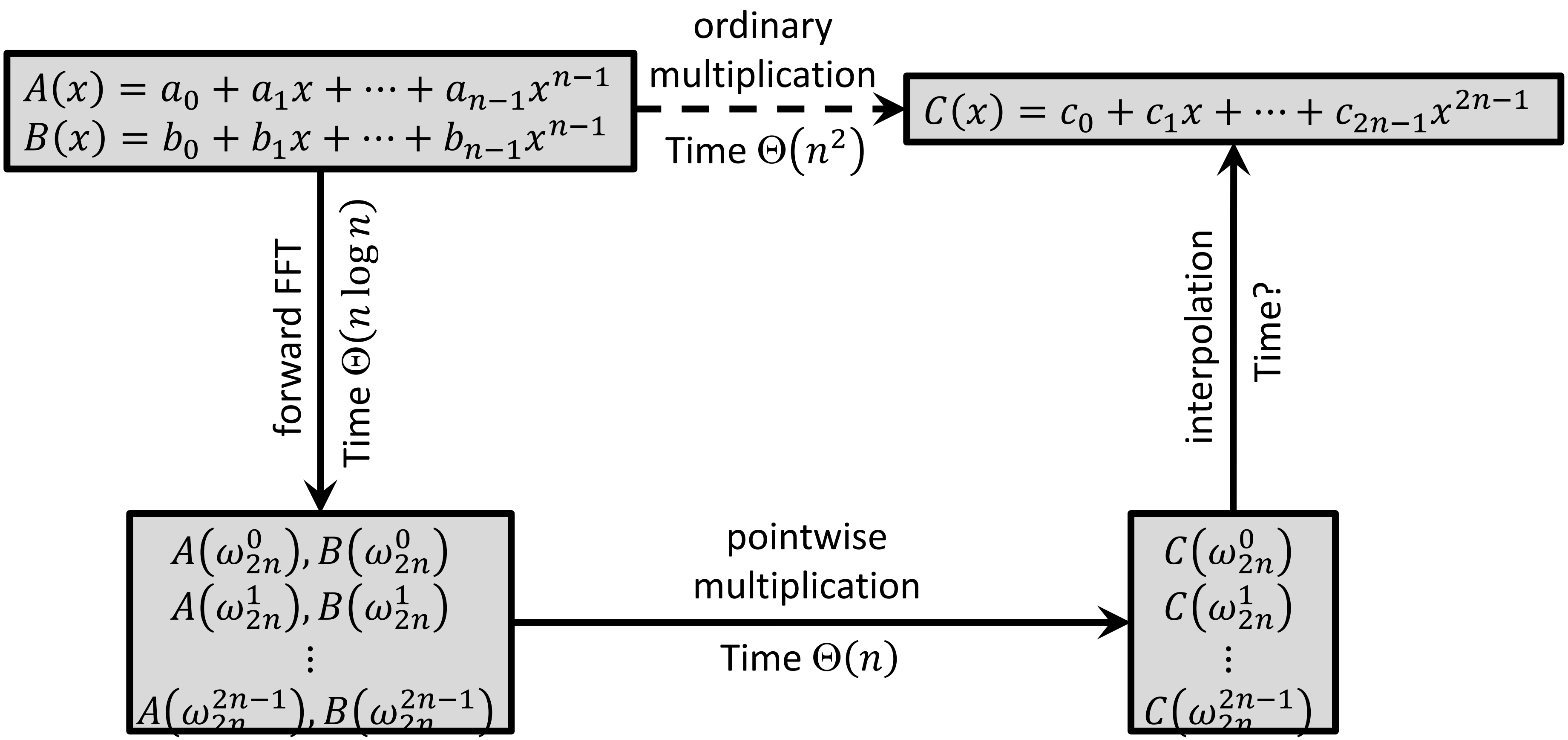
if $n = 1$ *then*
    return (a_0)
$\omega_n \leftarrow e^{2\pi i/n}$
$\omega \leftarrow 1$
$y^{even} \leftarrow$ *Rec-FFT* ($(a_0, a_2, \ldots, a_{n-2})$)
$y^{odd} \leftarrow$ *Rec-FFT* ($(a_1, a_3, \ldots, a_{n-1})$)
for $j \leftarrow 0$ *to* $n/2 - 1$ *do*
    $y_j \leftarrow y_j^{even} + \omega\, y_j^{odd}$
    $y_{n/2+j} \leftarrow y_j^{even} - \omega\, y_j^{odd}$
    $\omega \leftarrow \omega\, \omega_n$
return y

Running time:

$$T(n) = \begin{cases} \Theta(1), & if\ n = 1, \\ 2T\left(\frac{n}{2}\right) + \Theta(n), & otherwise. \end{cases}$$

$$= \Theta(n \log n)$$

Faster Polynomial Multiplication?
(in Coefficient Form)



Point-Value Form ⇒ Coefficient Form

Given: $$\underbrace{\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \omega_n & (\omega_n)^2 & \cdots & (\omega_n)^{n-1} \\ 1 & \omega_n^2 & \left(\omega_n^2\right)^2 & \cdots & \left(\omega_n^2\right)^{n-1} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ 1 & \omega_n^{n-1} & \left(\omega_n^{n-1}\right)^2 & \cdots & \left(\omega_n^{n-1}\right)^{n-1} \end{bmatrix}}_{\substack{V(\omega_n) \\ Vandermonde\ Matrix}} \underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_{n-1} \end{bmatrix}}_{\bar{a}} = \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_{n-1} \end{bmatrix}}_{\bar{y}}$$

$$\Rightarrow V(\omega_n) \cdot \bar{a} = \bar{y}$$

We want to solve: $\bar{a} = [V(\omega_n)]^{-1} \cdot \bar{y}$

It turns out that: $[V(\omega_n)]^{-1} = \frac{1}{n} V\left(\frac{1}{\omega_n}\right)$

That means $[V(\omega_n)]^{-1}$ looks almost similar to $V(\omega_n)$!

Point-Value Form ⇒ Coefficient Form

Show that: $[V(\omega_n)]^{-1} = \frac{1}{n} V\left(\frac{1}{\omega_n}\right)$

Let $U(\omega_n) = \frac{1}{n} V\left(\frac{1}{\omega_n}\right)$

We want to show that $U(\omega_n)V(\omega_n) = I_n$,
where I_n is the $n \times n$ identity matrix.

Observe that for $0 \le j, k \le n-1$, the $(j,k)^{th}$ entries are:

$$[V(\omega_n)]_{jk} = \omega_n^{jk} \quad \text{and} \quad [U(\omega_n)]_{jk} = \frac{1}{n}\omega_n^{-jk}$$

Then entry (p,q) of $U(\omega_n)V(\omega_n)$,

$$[U(\omega_n)V(\omega_n)]_{pq} = \sum_{k=0}^{n-1} [U(\omega_n)]_{pk} [V(\omega_n)]_{kq} = \frac{1}{n}\sum_{k=0}^{n-1} \omega_n^{k(q-p)}$$

Point-Value Form ⇒ Coefficient Form

$$[U(\omega_n)V(\omega_n)]_{pq} = \frac{1}{n}\sum_{k=0}^{n-1} \omega_n^{k(q-p)}$$

CASE $p = q$:

$$[U(\omega_n)V(\omega_n)]_{pq} = \frac{1}{n}\sum_{k=0}^{n-1} \omega_n^{0} = \frac{1}{n}\sum_{k=0}^{n-1} 1 = \frac{1}{n} \times n = 1$$

CASE $p \neq q$:

$$[U(\omega_n)V(\omega_n)]_{pq} = \frac{1}{n}\sum_{k=0}^{n-1} \left(\omega_n^{q-p}\right)^k = \frac{1}{n} \times \frac{\left(\omega_n^{q-p}\right)^n - 1}{\omega_n^{q-p} - 1}$$

$$= \frac{1}{n} \times \frac{(\omega_n^n)^{q-p} - 1}{\omega_n^{q-p} - 1} = \frac{1}{n} \times \frac{(1)^{q-p} - 1}{\omega_n^{q-p} - 1} = 0$$

Hence $U(\omega_n)V(\omega_n) = I_n$

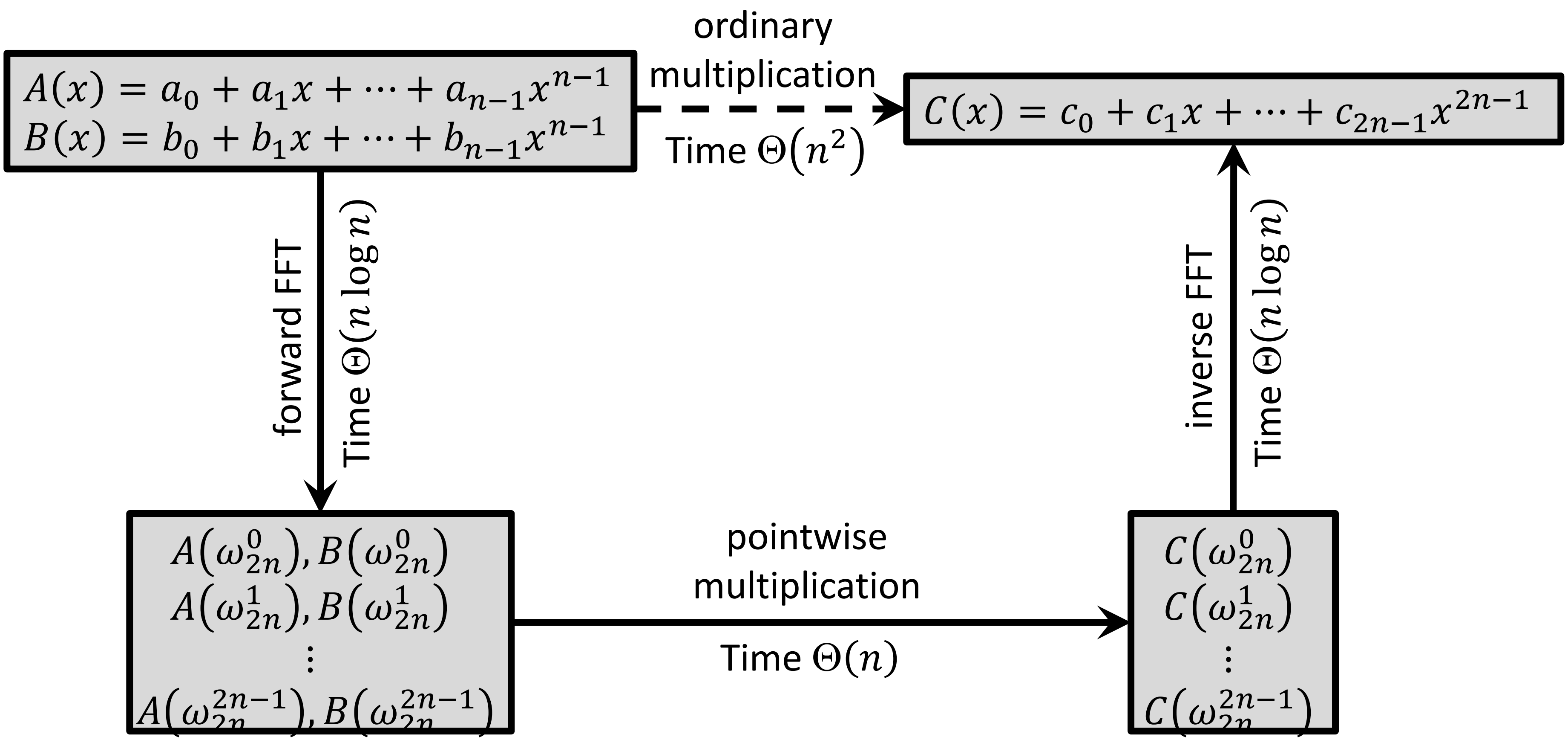
Point-Value Form ⇒ Coefficient Form

We need to compute the following matrix-vector product:

$$\underbrace{\begin{bmatrix} a_0 \\ a_1 \\ a_2 \\ \cdot \\ \cdot \\ a_{n-1} \end{bmatrix}}_{\bar{a}} = \frac{1}{n} \times \underbrace{\begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ 1 & \frac{1}{\omega_n} & \left(\frac{1}{\omega_n}\right)^2 & \cdots & \left(\frac{1}{\omega_n}\right)^{n-1} \\ 1 & \frac{1}{\omega_n^2} & \left(\frac{1}{\omega_n^2}\right)^2 & \cdots & \left(\frac{1}{\omega_n^2}\right)^{n-1} \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ \cdot & \cdot & \cdot & \cdots & \cdot \\ 1 & \frac{1}{\omega_n^{n-1}} & \left(\frac{1}{\omega_n^{n-1}}\right)^2 & \cdots & \left(\frac{1}{\omega_n^{n-1}}\right)^{n-1} \end{bmatrix}}_{[V(\omega_n)]^{-1}} \underbrace{\begin{bmatrix} y_0 \\ y_1 \\ y_2 \\ \cdot \\ \cdot \\ y_{n-1} \end{bmatrix}}_{\bar{y}}$$

This inverse problem is almost similar to the forward problem, and can be solved in $\Theta(n \log n)$ time using the same algorithm as the forward FFT with only minor modifications!

Faster Polynomial Multiplication? (in Coefficient Form)



Two polynomials of degree bound n given in the coefficient form can be multiplied in $\Theta(n \log n)$ time!

Some Applications of Fourier Transform and FFT

- Signal processing
- Image processing
- Noise reduction
- Data compression
- Solving partial differential equation
- Multiplication of large integers
- Polynomial multiplication
- Molecular docking

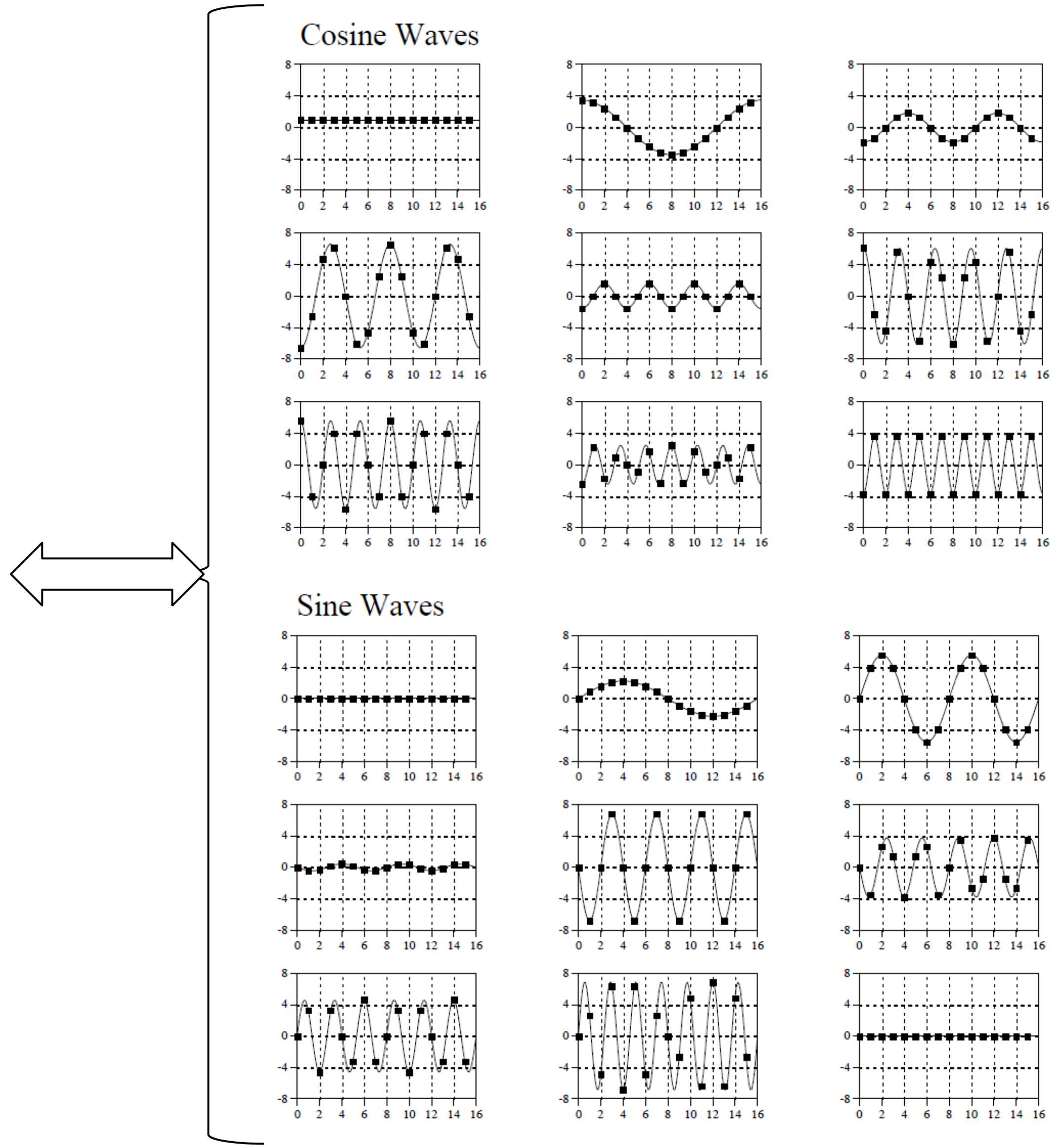
Some Applications of Fourier Transform and FFT

Jean Baptiste Joseph Fourier

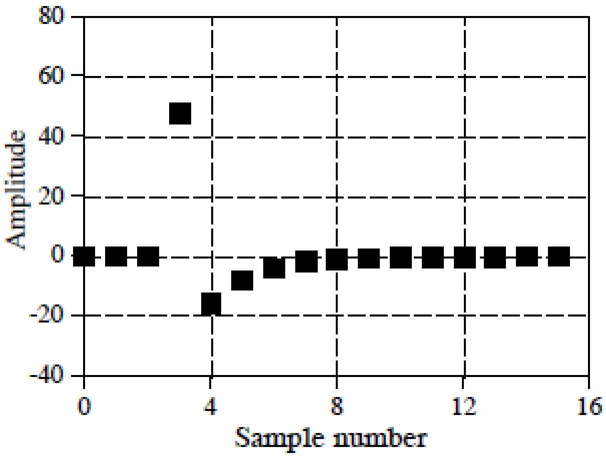
Any periodic signal can be represented as a sum of a series of sinusoidal (sine & cosine) waves. [1807]

Spatial (Time) Domain ⇔ Frequency Domain

Frequency Domain



Spatial (Time) Domain



Source: The Scientist and Engineer's Guide to Digital Signal Processing by Steven W. Smith

Spatial (Time) Domain ⇔ Frequency Domain

Function *s*(*x*) (in red) is a sum of six sine functions of different amplitudes and harmonically related frequencies. The Fourier transform, *S*(*f*) (in blue), which depicts amplitude vs frequency, reveals the 6 frequencies and their amplitudes.

Source: http://en.wikipedia.org/wiki/Fourier_series#mediaviewer/File:Fourier_series_and_transform.gif (uploaded by Bob K.)

Spatial (Time) Domain ⇔ Frequency Domain

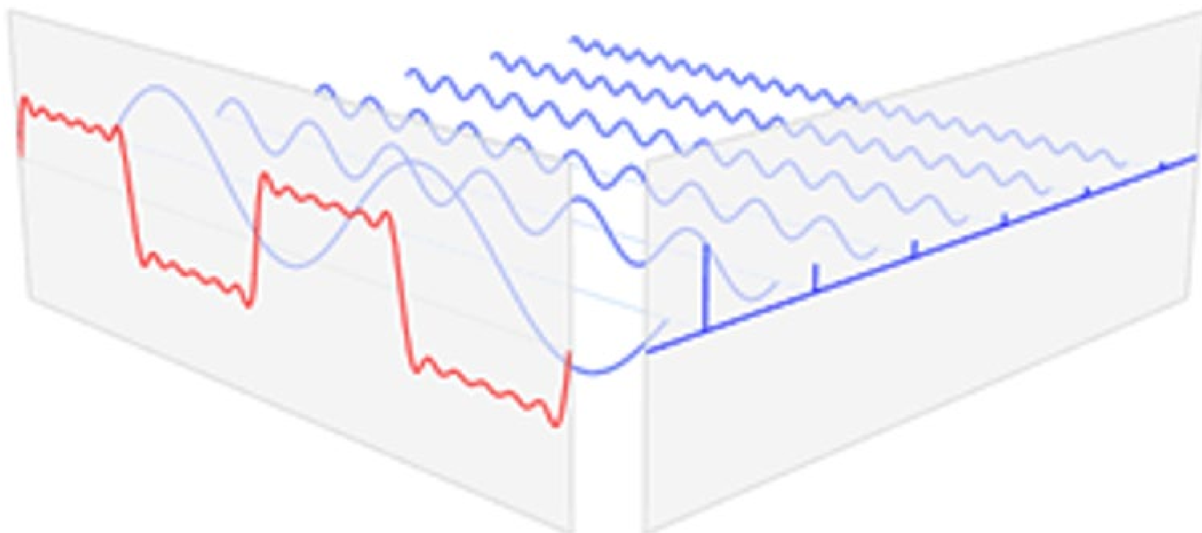
$$s_6(x)$$

$$a_n \cos(nx) + b_n \sin(nx)$$

Function $s(x)$ (in red) is a sum of six sine functions of different amplitudes and harmonically related frequencies. The Fourier transform, $S(f)$ (in blue), which depicts amplitude vs frequency, reveals the 6 frequencies and their amplitudes.

Source: http://en.wikipedia.org/wiki/Fourier_series#mediaviewer/File:Fourier_series_and_transform.gif (uploaded by Bob K.)

Spatial (Time) Domain ⇔ Frequency Domain



Function *s*(*x*) (in red) is a sum of six sine functions of different amplitudes and harmonically related frequencies. The Fourier transform, *S*(*f*) (in blue), which depicts amplitude vs frequency, reveals the 6 frequencies and their amplitudes.

Source: http://en.wikipedia.org/wiki/Fourier_series#mediaviewer/File:Fourier_series_and_transform.gif (uploaded by Bob K.)

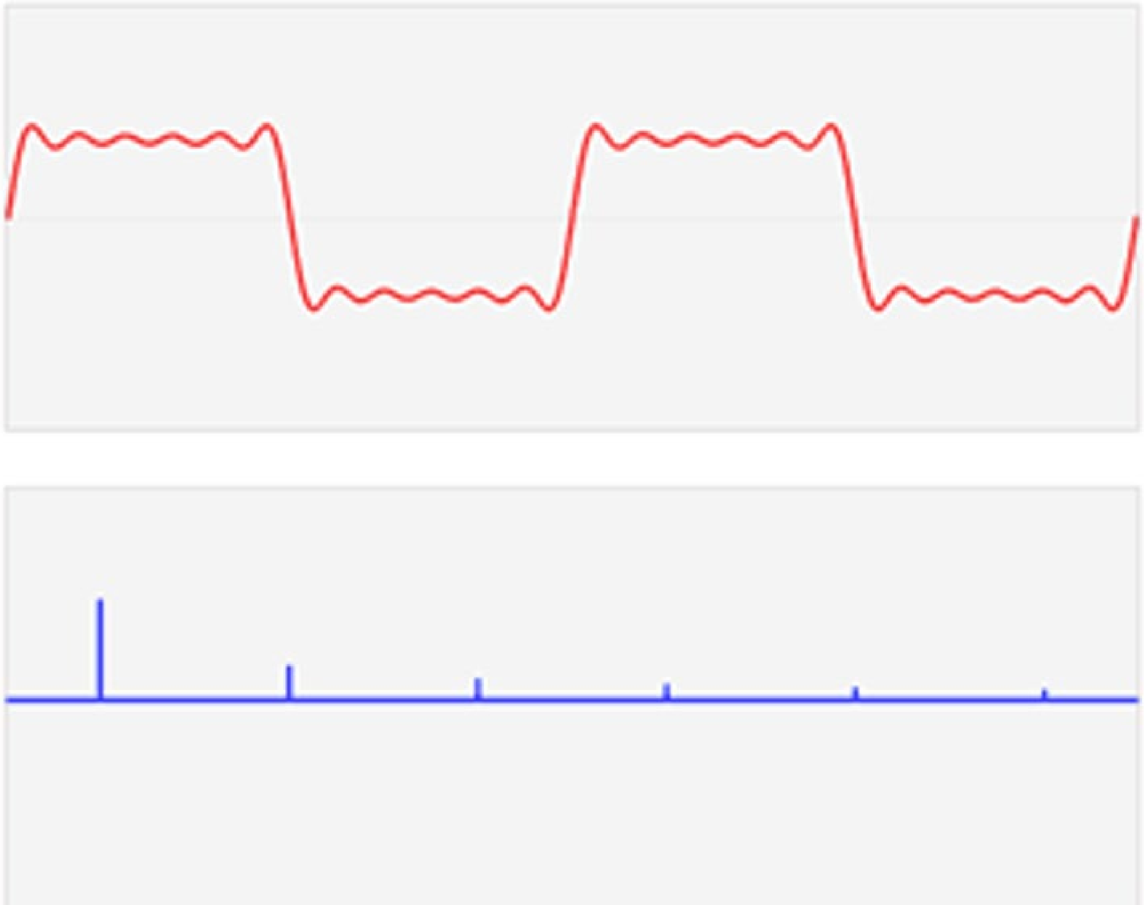
Spatial (Time) Domain ⇔ Frequency Domain

$S(f)$

Function $s(x)$ (in red) is a sum of six sine functions of different amplitudes and harmonically related frequencies. The Fourier transform, $S(f)$ (in blue), which depicts amplitude vs frequency, reveals the 6 frequencies and their amplitudes.

Source: http://en.wikipedia.org/wiki/Fourier_series#mediaviewer/File:Fourier_series_and_transform.gif (uploaded by Bob K.)

Spatial (Time) Domain ⇔ Frequency Domain



Function $s(x)$ (in red) is a sum of six sine functions of different amplitudes and harmonically related frequencies. The Fourier transform, $S(f)$ (in blue), which depicts amplitude vs frequency, reveals the 6 frequencies and their amplitudes.

Source: http://en.wikipedia.org/wiki/Fourier_series#mediaviewer/File:Fourier_series_and_transform.gif (uploaded by Bob K.)

Spatial (Time) Domain ⇔ Frequency Domain (Fourier Transforms)

Let $s(t)$ be a signal specified in the time domain.

The strength of $s(t)$ at frequency f is given by:

$$S(f) = \int_{-\infty}^{\infty} s(t) \cdot e^{-2\pi i f t} \, dt$$

Evaluating this integral for all values of f gives the frequency domain function.

Now $s(t)$ can be retrieved by summing up the signal strengths at all possible frequencies:

$$s(t) = \int_{-\infty}^{\infty} S(f) \cdot e^{2\pi i f t} \, df$$

Why do the Transforms Work?

Let's try to get a little intuition behind why the transforms work.

We will look at a very simple example.

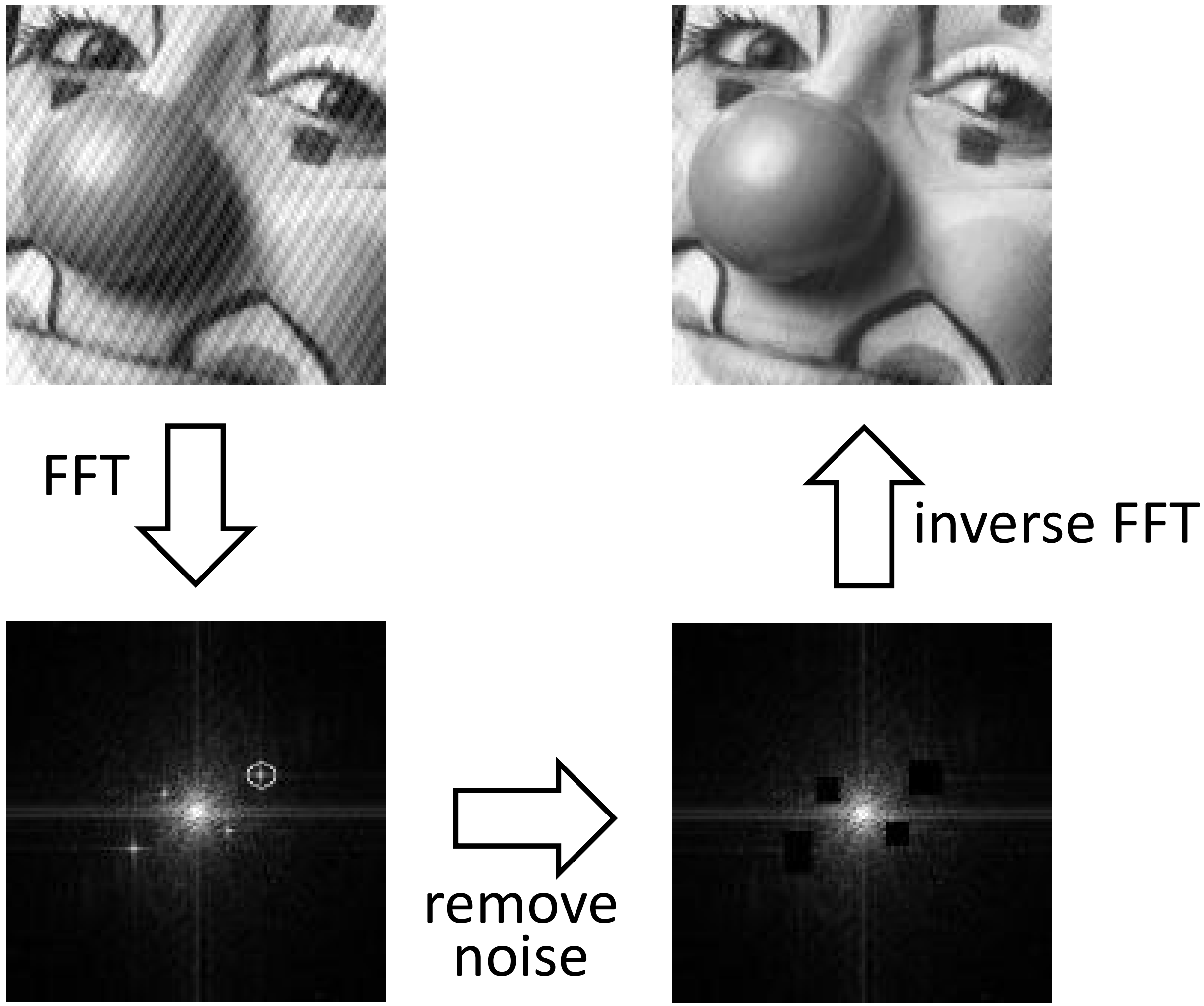
Suppose: $s(t) = \cos(2\pi h \cdot t)$

$$\frac{1}{T}\int_{-T}^{T} s(t) \cdot e^{-2\pi i f t}\,dt = \begin{cases} 1 + \frac{\sin(4\pi f T)}{4\pi f T}, & \text{if } f = h, \\ \frac{\sin(2\pi(h-f)T)}{2\pi(h-f)T} + \frac{\sin(2\pi(h+f)T)}{2\pi(h+f)T}, & \text{otherwise.} \end{cases}$$

$$\Rightarrow \lim_{T\to\infty}\left(\frac{1}{T}\int_{-T}^{T} s(t) \cdot e^{-2\pi i f t}\,dt\right) = \begin{cases} 1, & \text{if } f = h, \\ 0, & \text{otherwise.} \end{cases}$$

So, the transform can detect if $f = h$!

Noise Reduction



Source: http://www.mediacy.com/index.aspx?page=AH_FFTExample

Data Compression

- Discrete Cosine Transforms (DCT) are used for lossy data compression (e.g., MP3, JPEG, MPEG)
- DCT is a Fourier-related transform similar to DFT (Discrete Fourier Transform) but uses only real data (uses cosine waves only instead of both cosine and sine waves)
- Forward DCT transforms data from spatial to frequency domain
- Each frequency component is represented using a fewer number of bits (i.e., truncated / quantized)
- Low amplitude high frequency components are also removed
- Inverse DCT then transforms the data back to spatial domain
- The resulting image compresses better

Data Compression

Transformation to frequency domain using cosine transforms work in the same way as the Fourier transform.

Suppose: $s(t) = \cos(2\pi h \cdot t)$

$$\frac{1}{T}\int_{-T}^{T} s(t) \cdot \cos(2\pi f t)\, dt = \begin{cases} 1 + \dfrac{\sin(4\pi f T)}{4\pi f T}, & \text{if } f = h, \\[2ex] \dfrac{\sin(2\pi(h-f)T)}{2\pi(h-f)T} + \dfrac{\sin(2\pi(h+f)T)}{2\pi(h+f)T}, & \text{otherwise.} \end{cases}$$

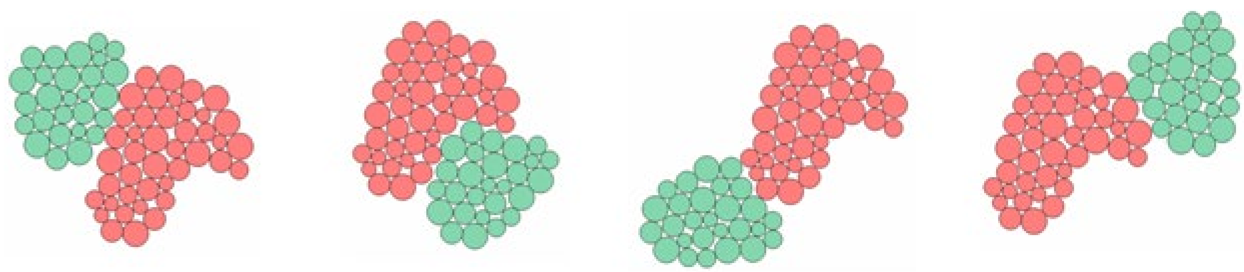
$$\Rightarrow \lim_{T\to\infty}\left(\frac{1}{T}\int_{-T}^{T} s(t) \cdot \cos(2\pi f t)\, dt\right) = \begin{cases} 1, & \text{if } f = h, \\ 0, & \text{otherwise.} \end{cases}$$

So, this transform can also detect if $f = h$.

Protein-Protein Docking

- Knowledge of complexes is used in
  - Drug design
  - Studying molecular assemblies
  - Structure function analysis
  - Protein interactions

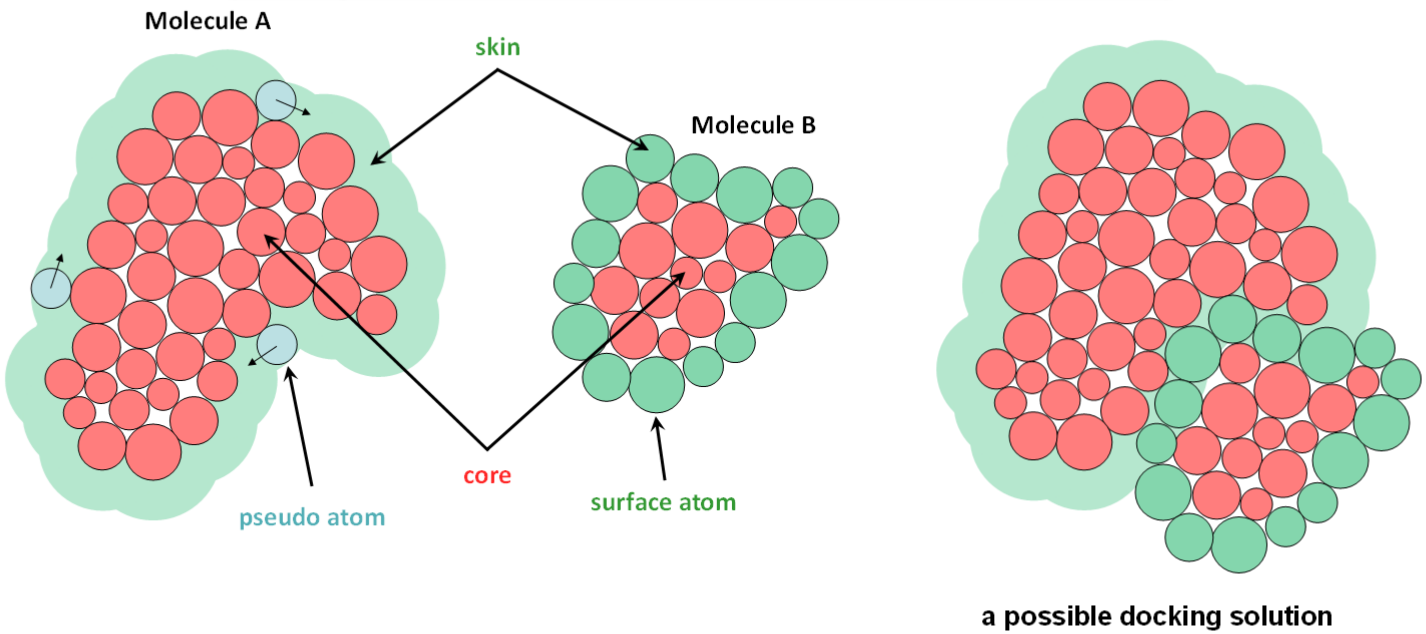
- Protein-Protein Docking: Given two proteins, find the best relative transformation and conformations to obtain a stable complex.



- Docking is a hard problem
  - Search space is huge (6D for rigid proteins)
  - Protein flexibility adds to the difficulty

Shape Complementarity

[Wang'91, Katchalski-Katzir et al.'92, Chen et al.'03]



To maximize skin-skin overlaps and minimize core-core overlaps

- assign positive real weights to skin atoms
- assign positive imaginary weights to core atoms

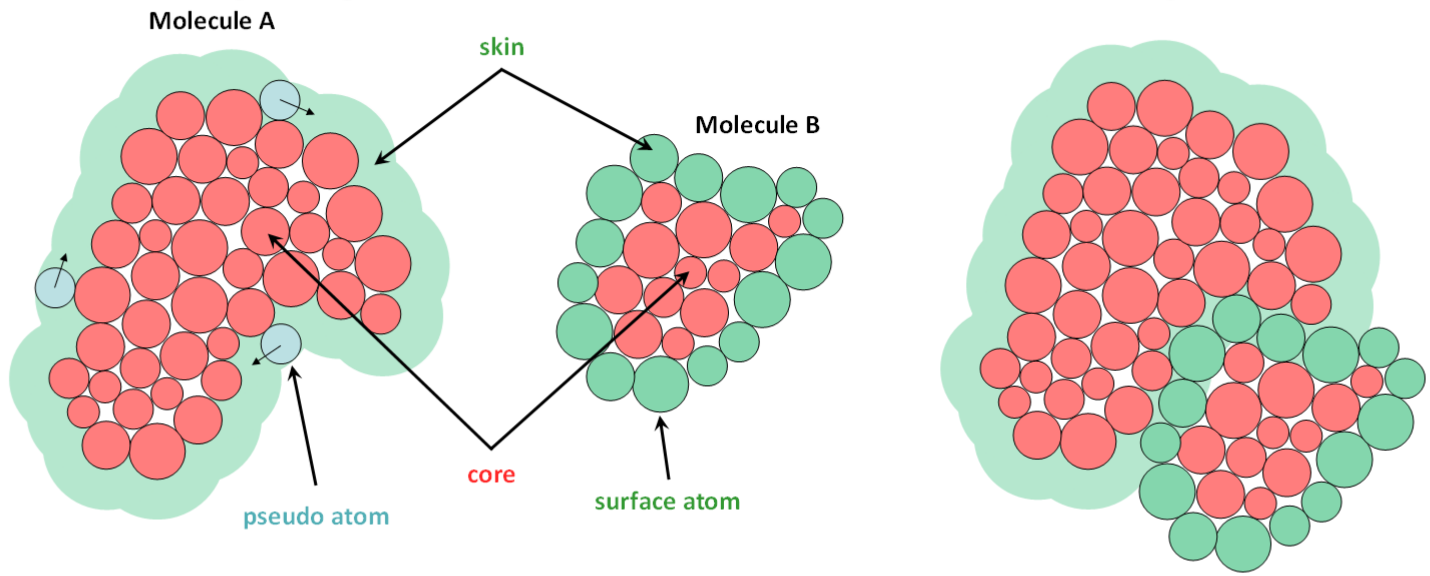
Let A' denote molecule A with the pseudo skin atoms.

For $P \in \{A', B\}$ with M_P atoms, affinity function: $f_P(x) = \sum_{k=1}^{M_P} w_k \cdot g_k(x)$

Here $g_k(x)$ is a Gaussian representation of atom k, and w_k its weight.

Shape Complementarity

[Wang'91, Katchalski-Katzir et al.'92, Chen et al.'03]



a possible docking solution

Let A' denote molecule A with the pseudo skin atoms.

For $P \in \{A', B\}$ with M_P atoms, affinity function:

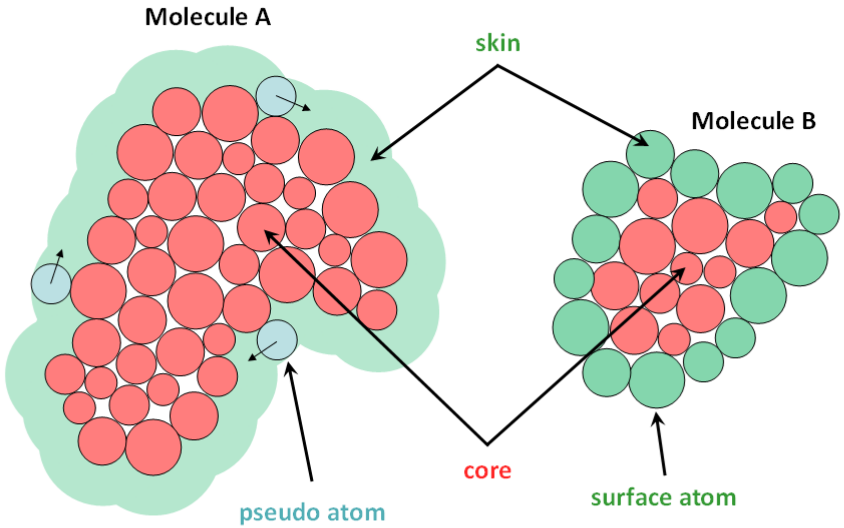
$$f_P(x) = \sum_{k=1}^{M_P} w_k \cdot g_k(x)$$

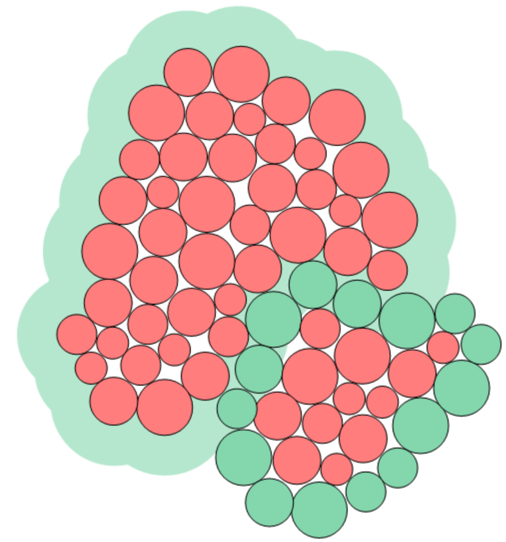
For rotation r and translation t of molecule B (i.e., $B_{t,r}$),

the interaction score, $F_{A,B}(t,r) = \int_x \; f_{A'}(x) f_{B_{t,r}}(x)\, dx$

Shape Complementarity

[Wang'91, Katchalski-Katzir et al.'92, Chen et al.'03]





a possible docking solution

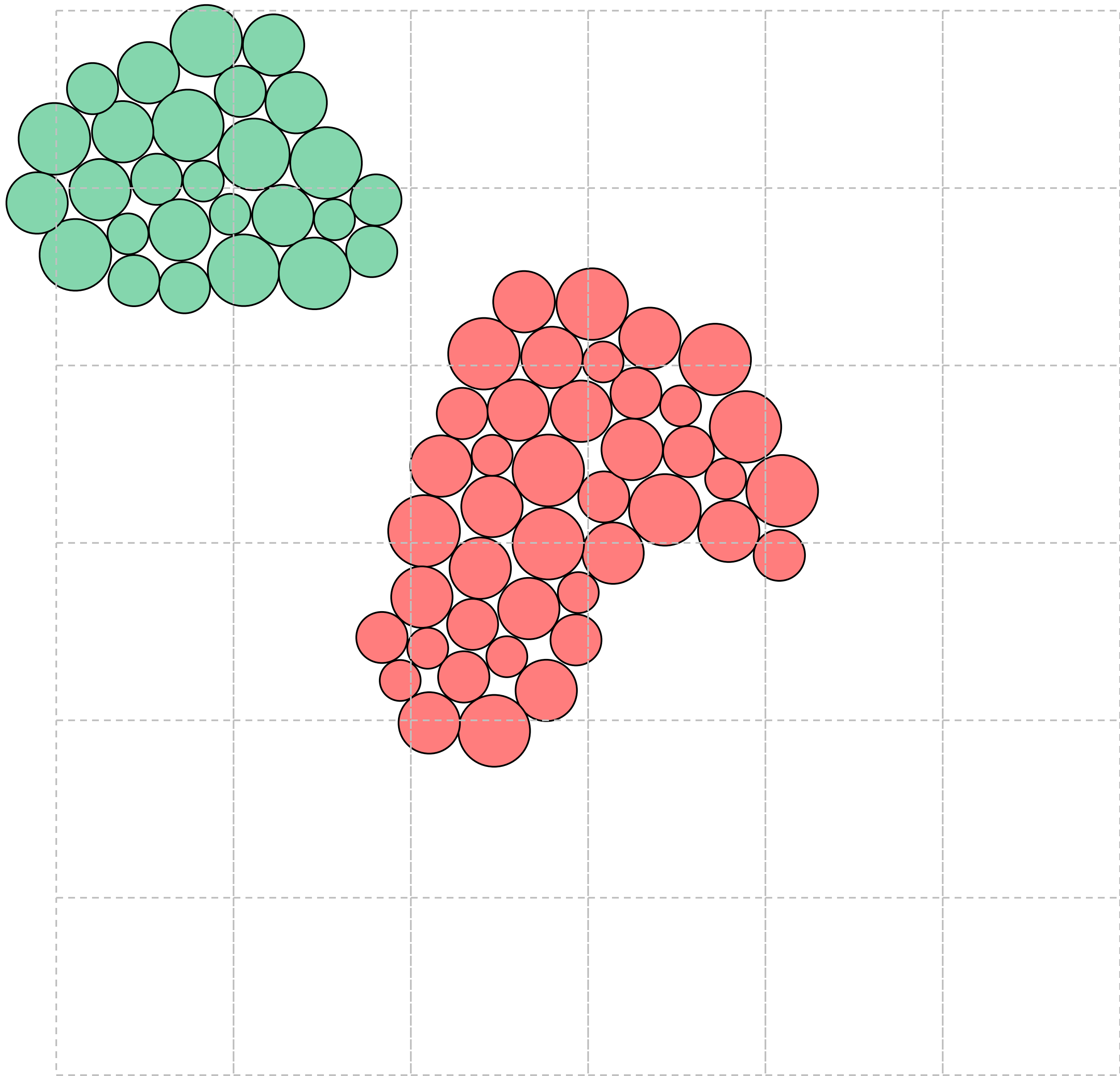
For rotation r and translation t of molecule B (i.e., $B_{t,r}$),

the interaction score, $F_{A,B}(t,r) = \int_x \; f_{A'}(x) f_{B_{t,r}}(x)\, dx$

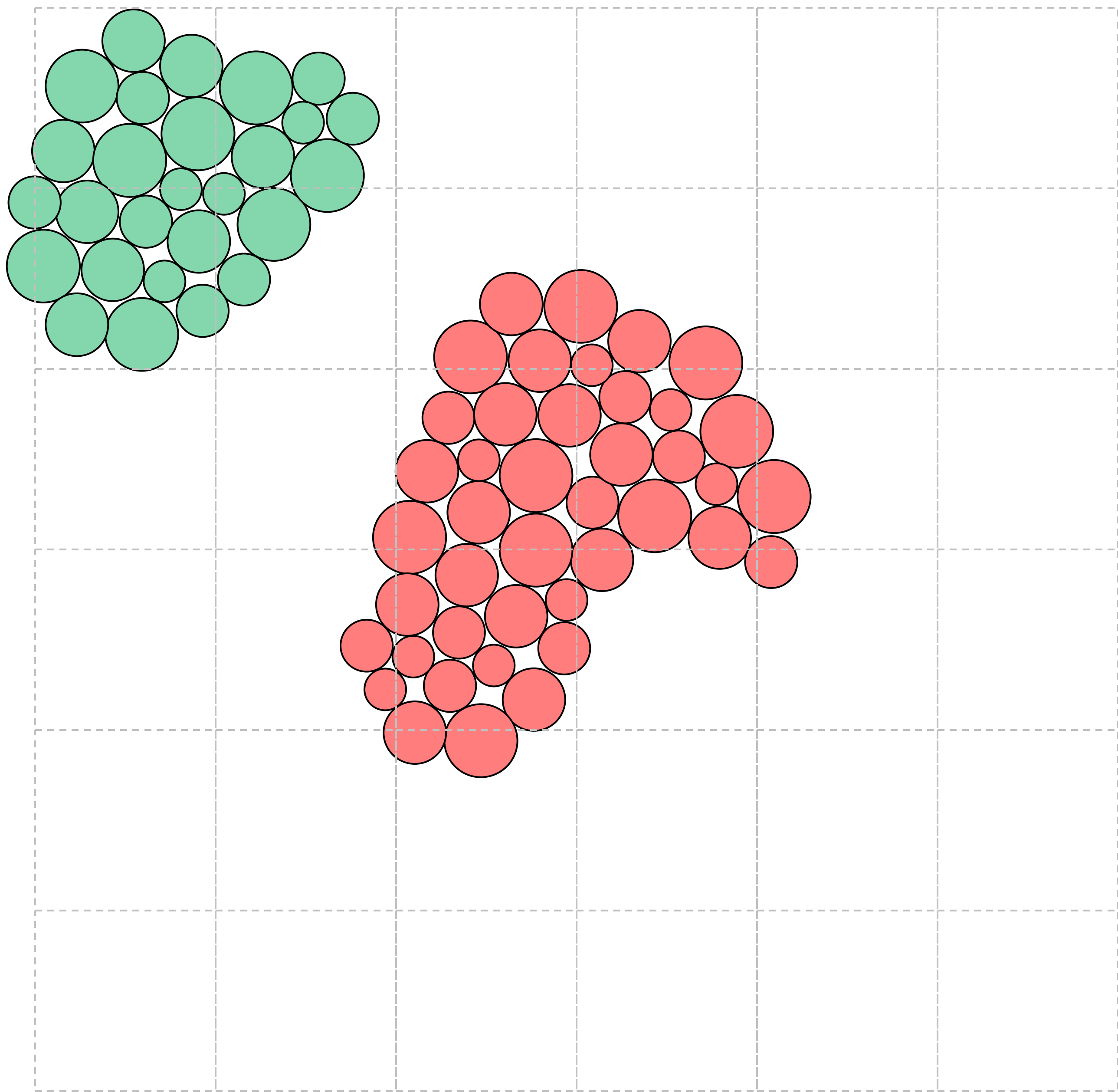
$$Re\left(F_{A,B}(t,r)\right) = \text{skin-skin overlap score} - \text{core-core overlap score}$$

$$Im\left(F_{A,B}(t,r)\right) = \text{skin-core overlap score}$$

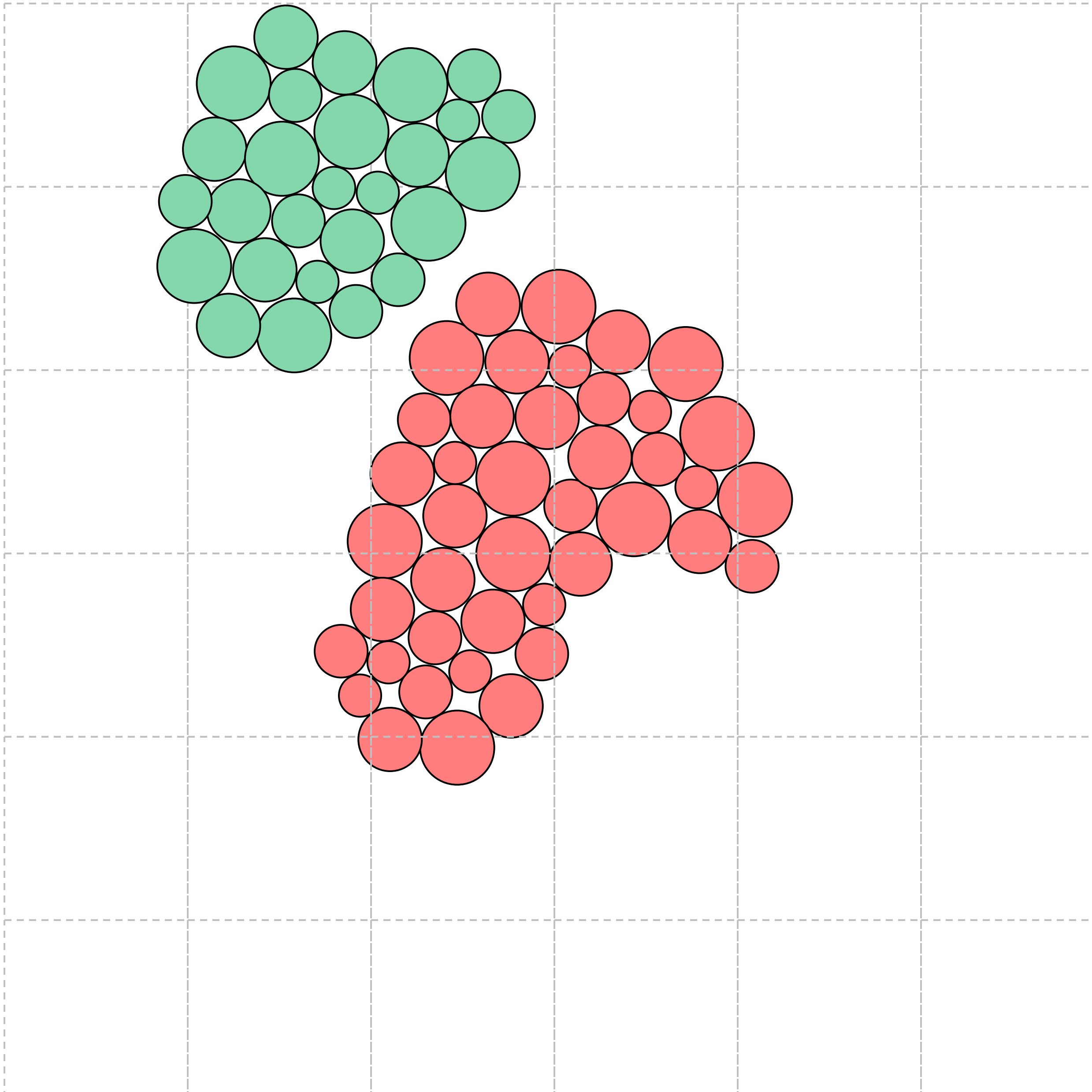
Docking: Rotational & Translational Search



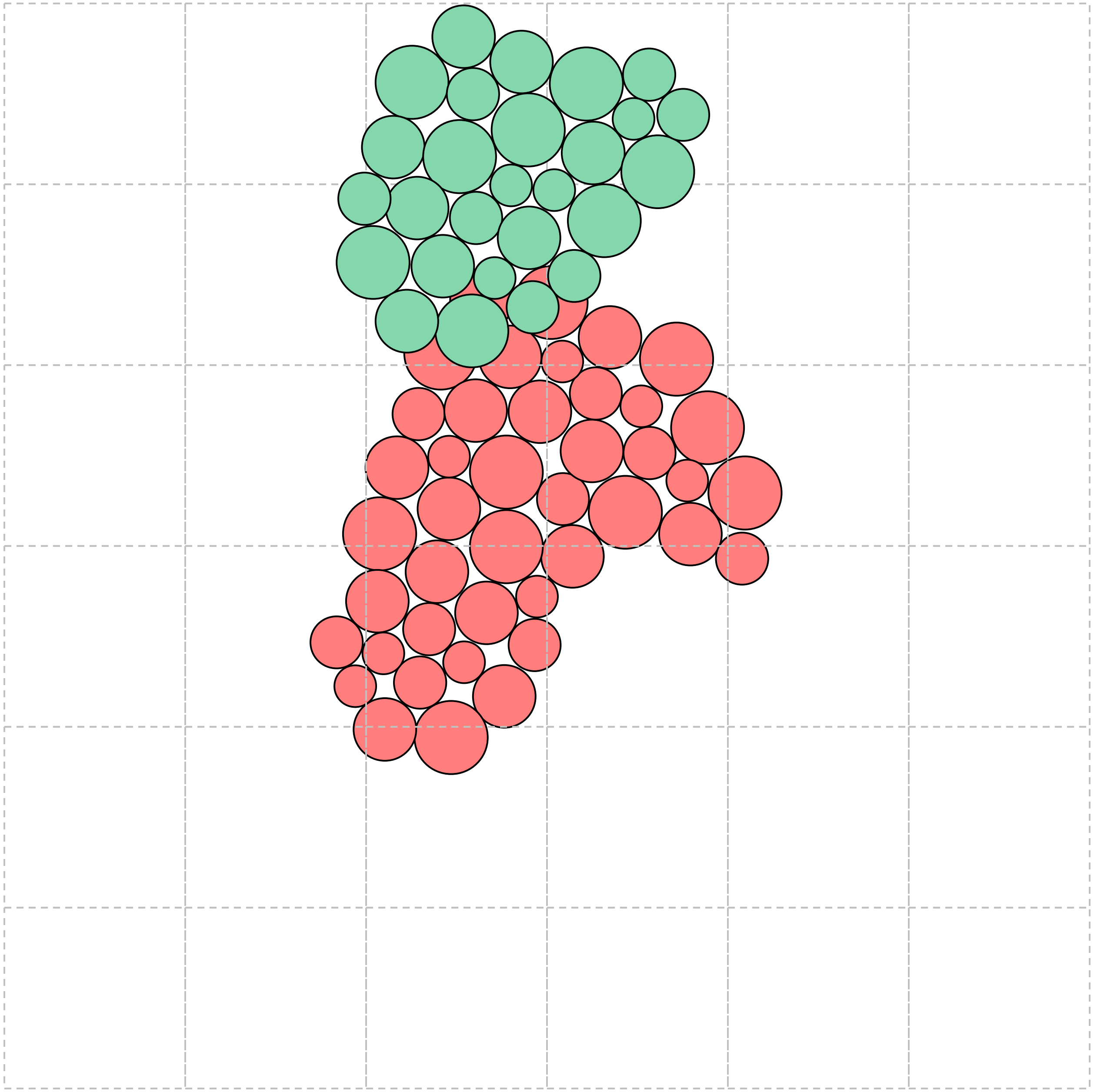
Docking: Rotational & Translational Search



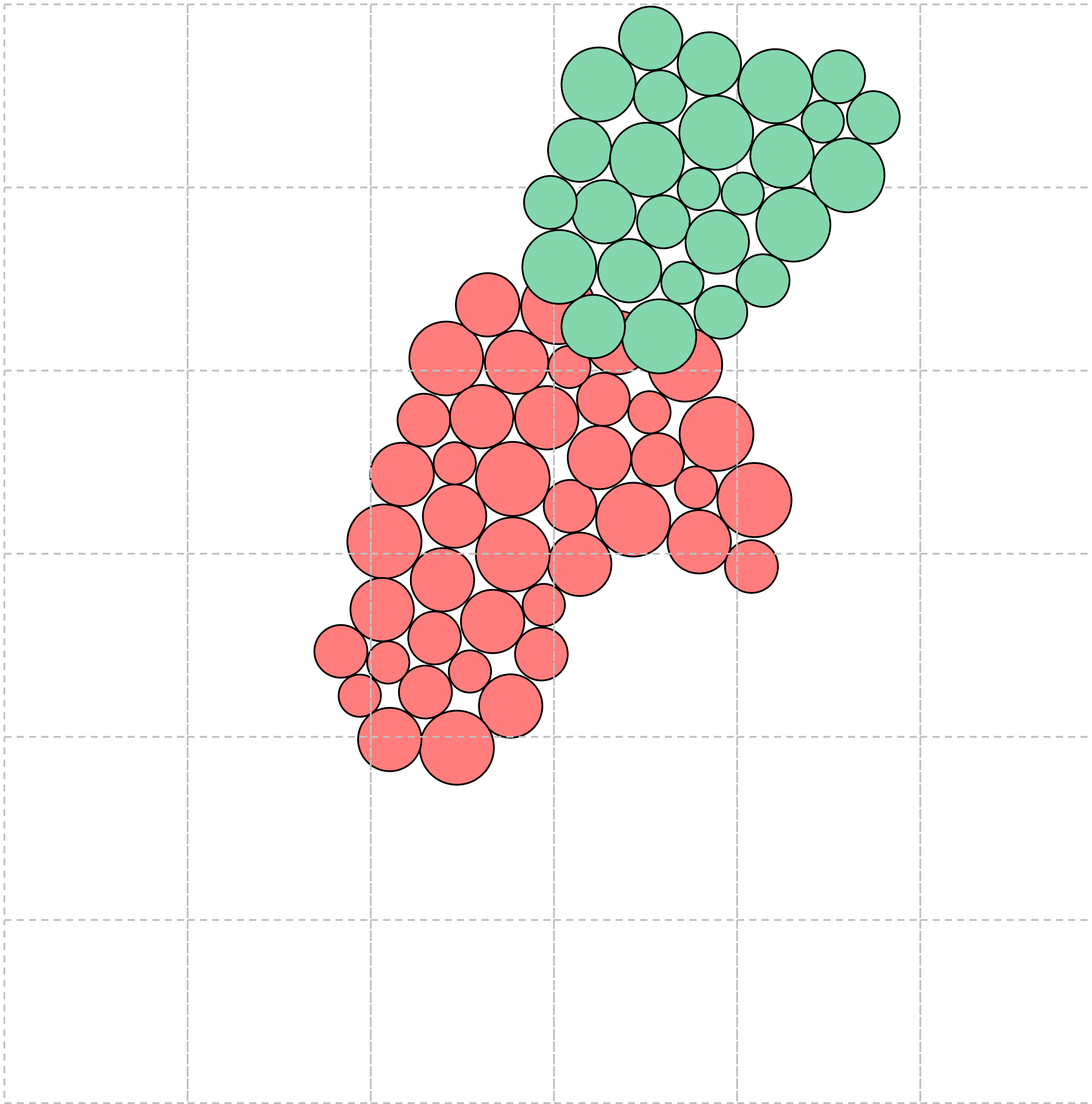
Docking: Rotational & Translational Search



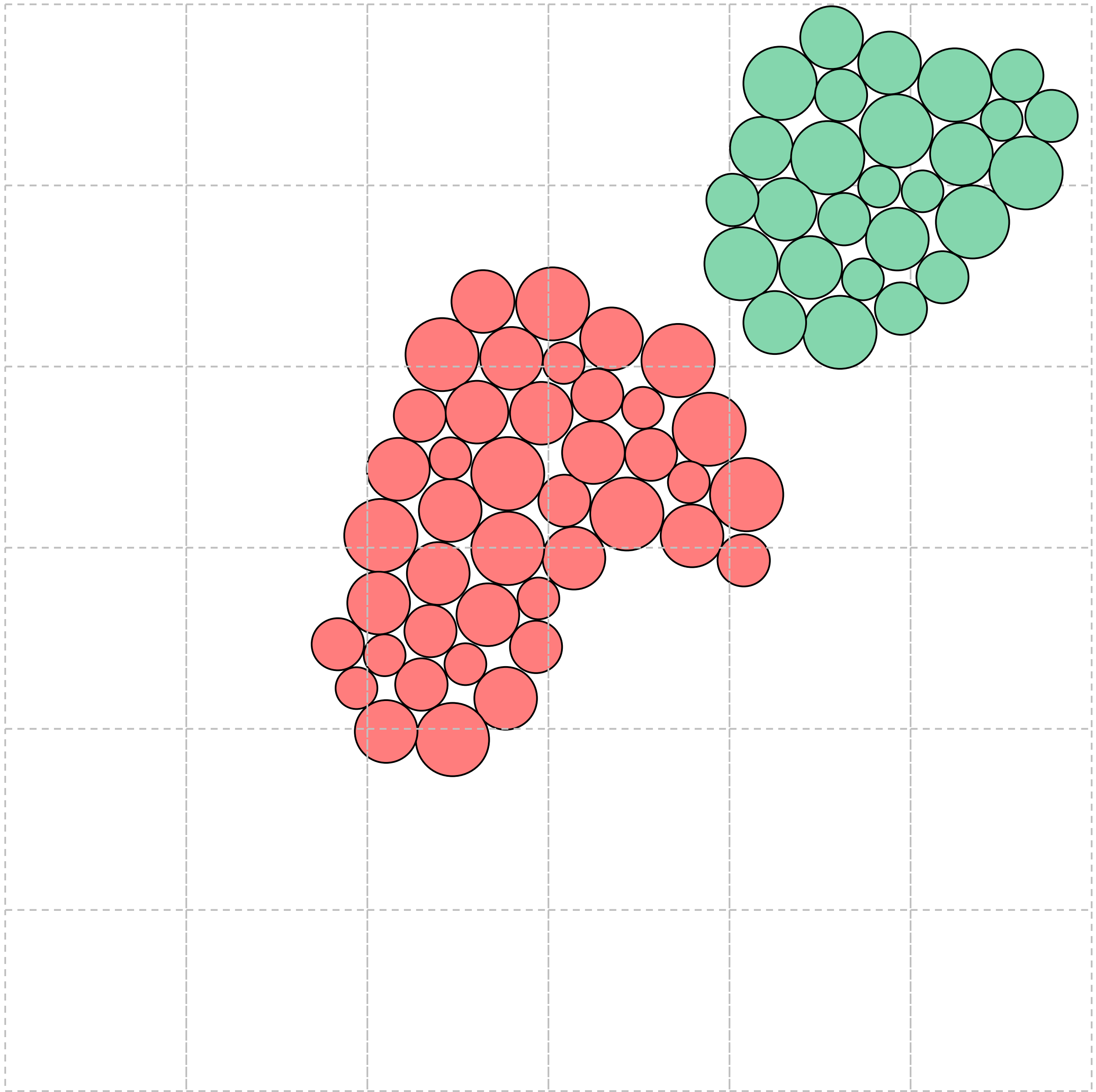
Docking: Rotational & Translational Search



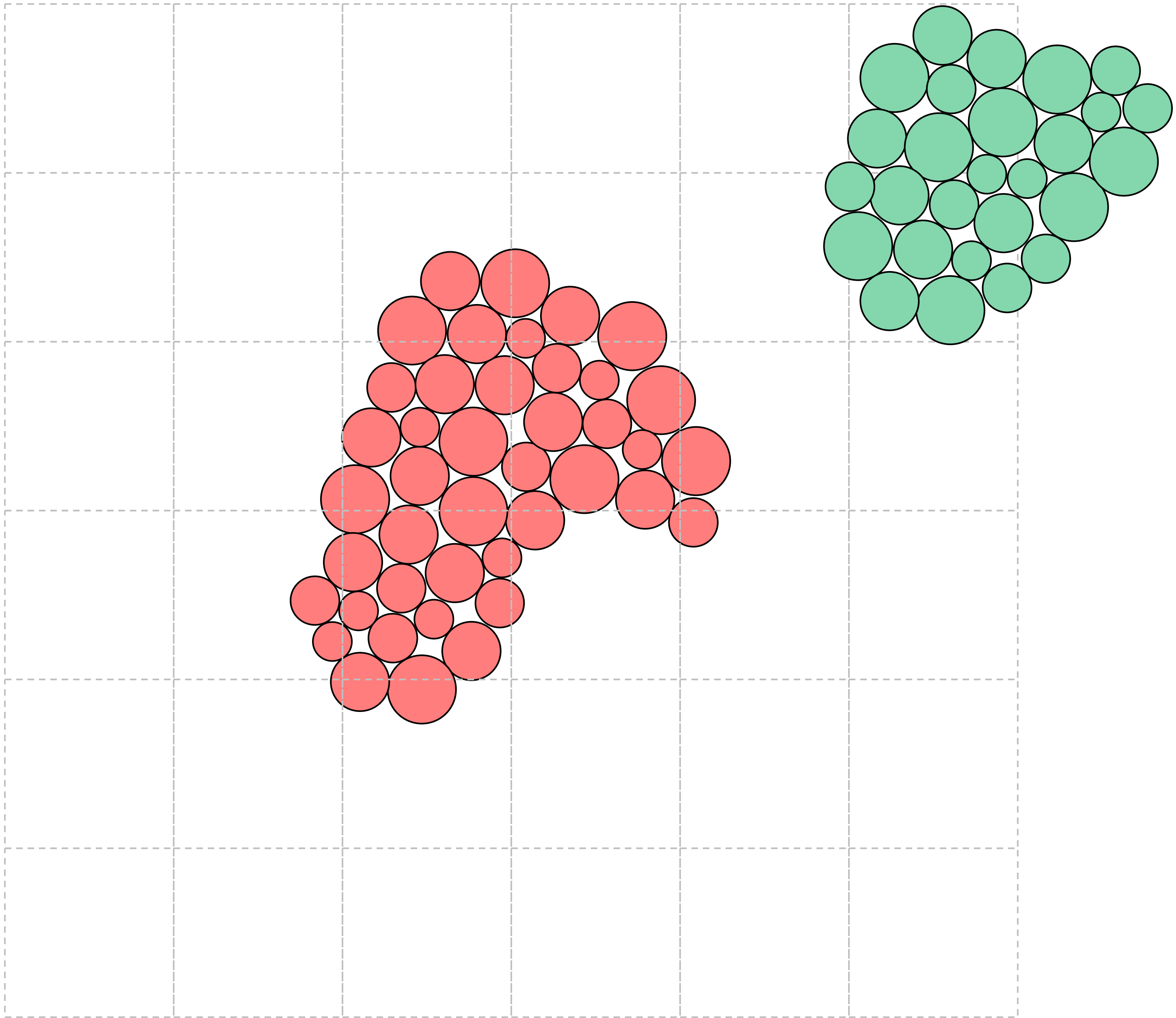
Docking: Rotational & Translational Search



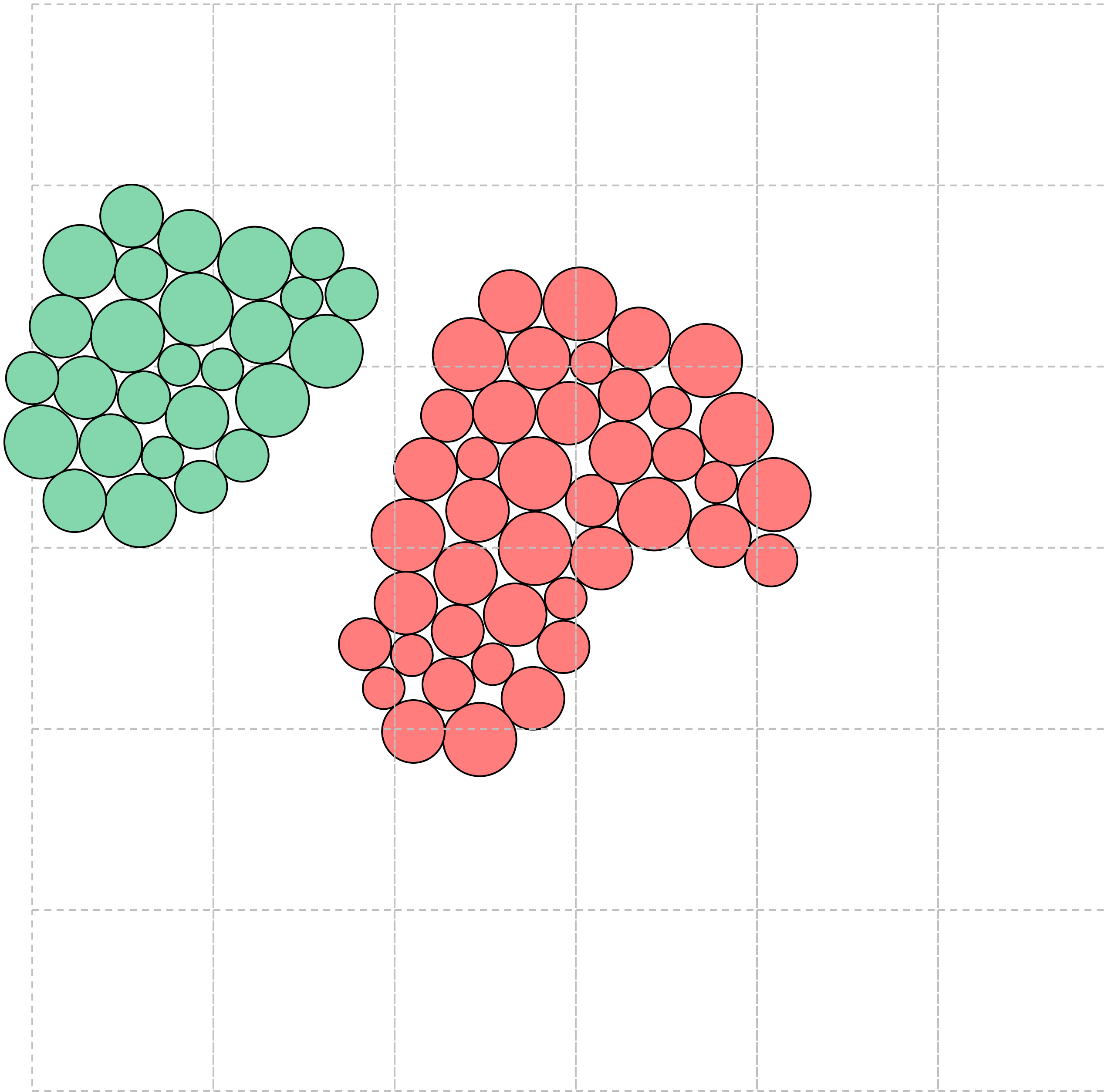
Docking: Rotational & Translational Search



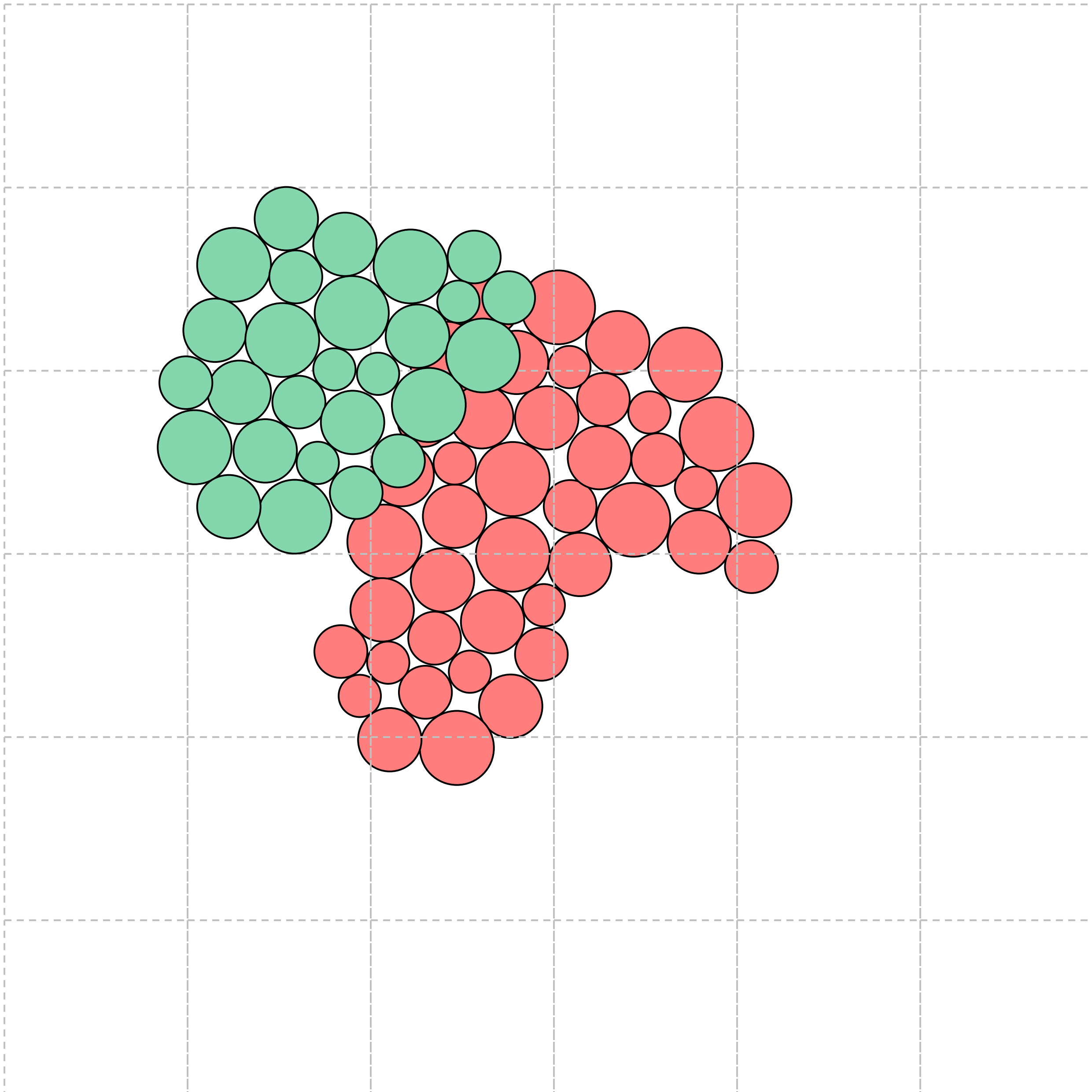
Docking: Rotational & Translational Search



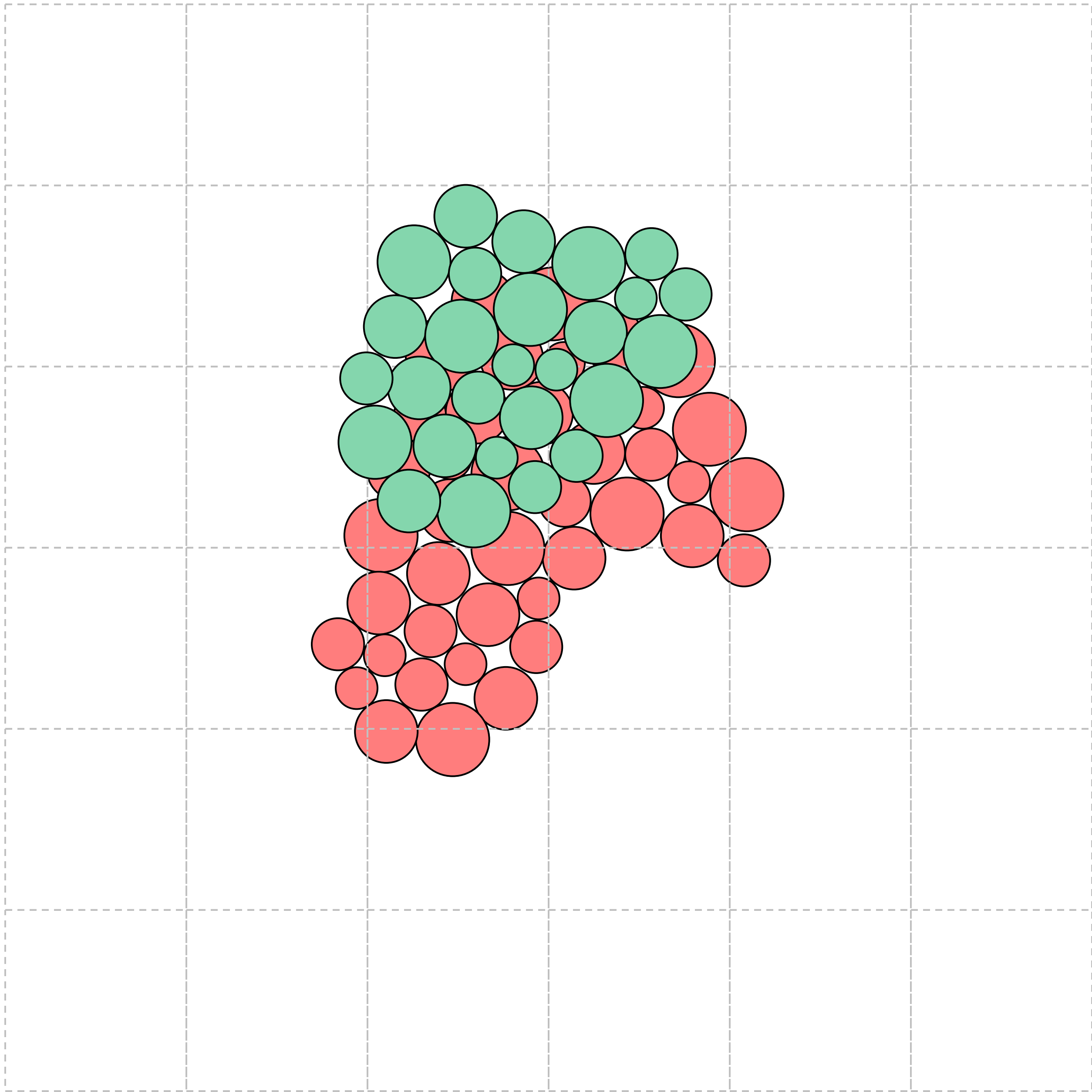
Docking: Rotational & Translational Search



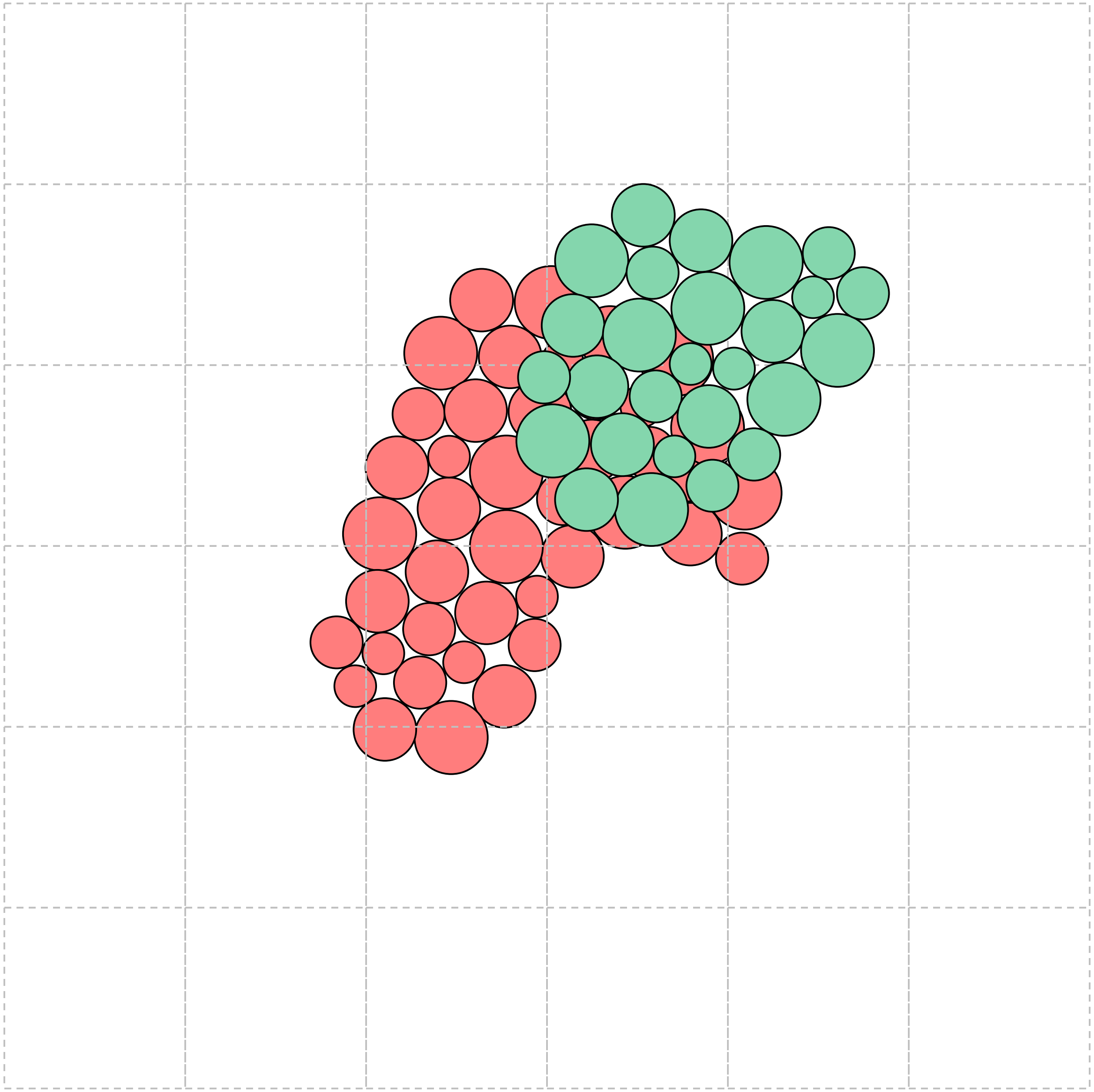
Docking: Rotational & Translational Search



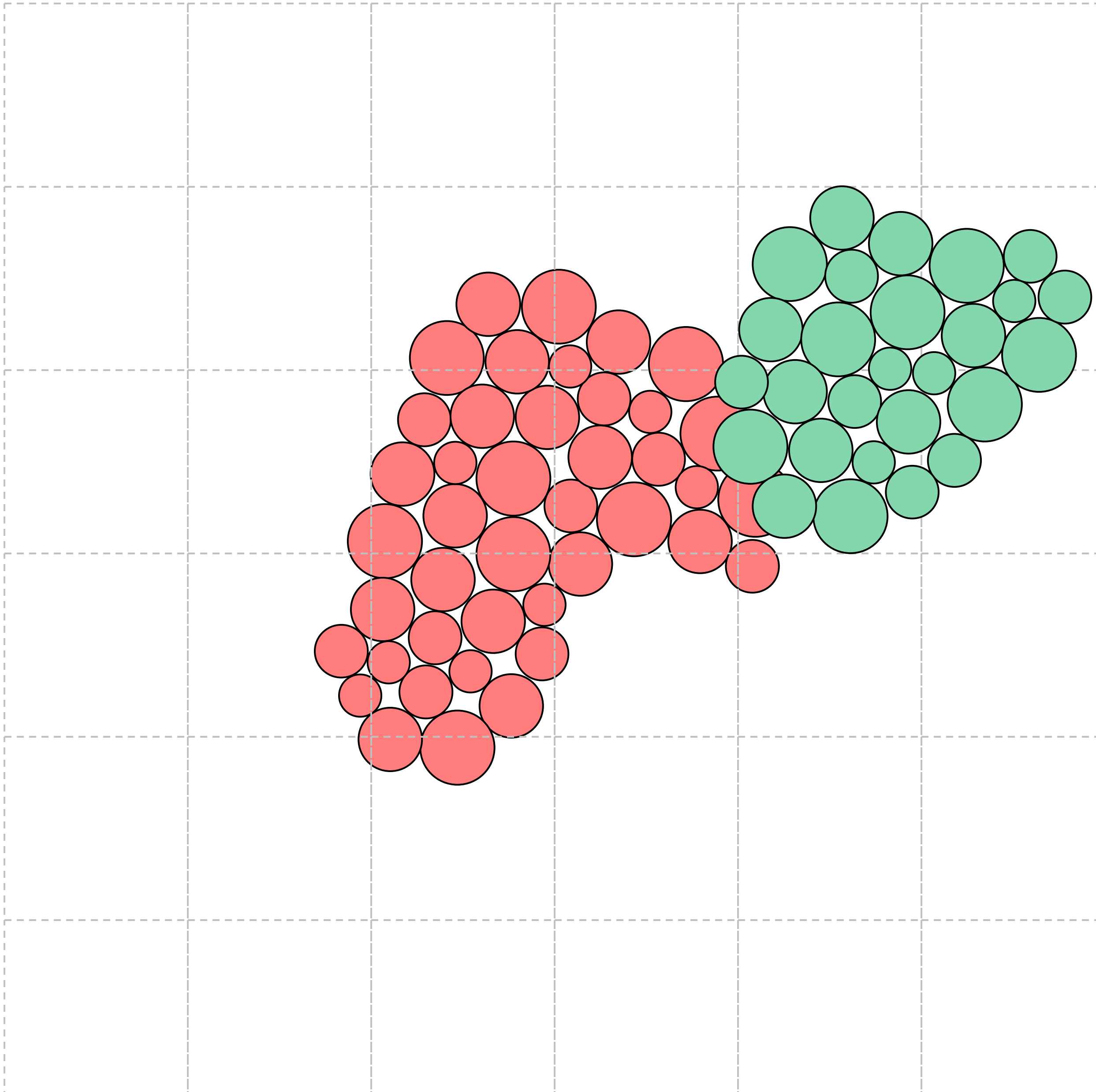
Docking: Rotational & Translational Search



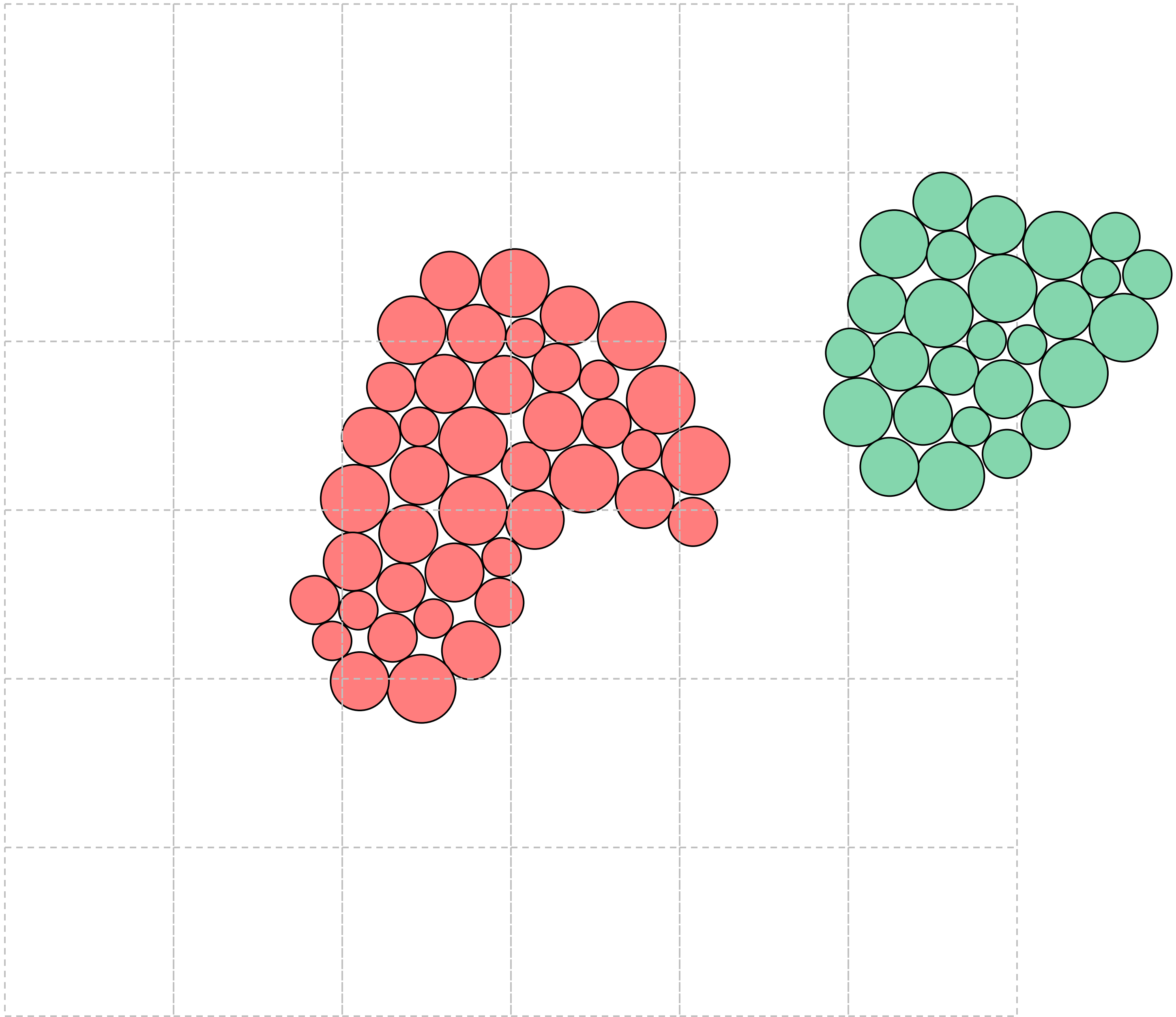
Docking: Rotational & Translational Search



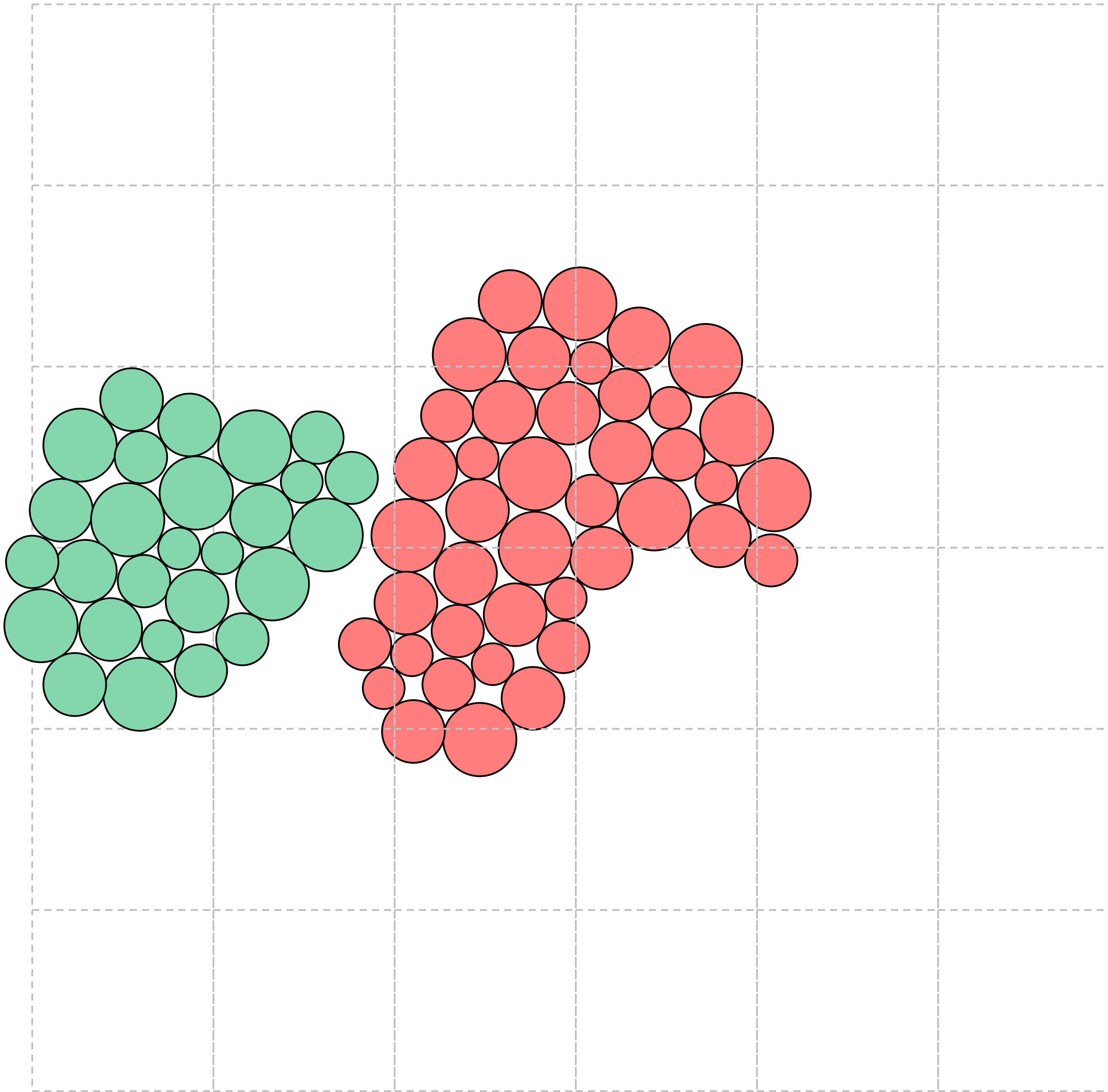
Docking: Rotational & Translational Search



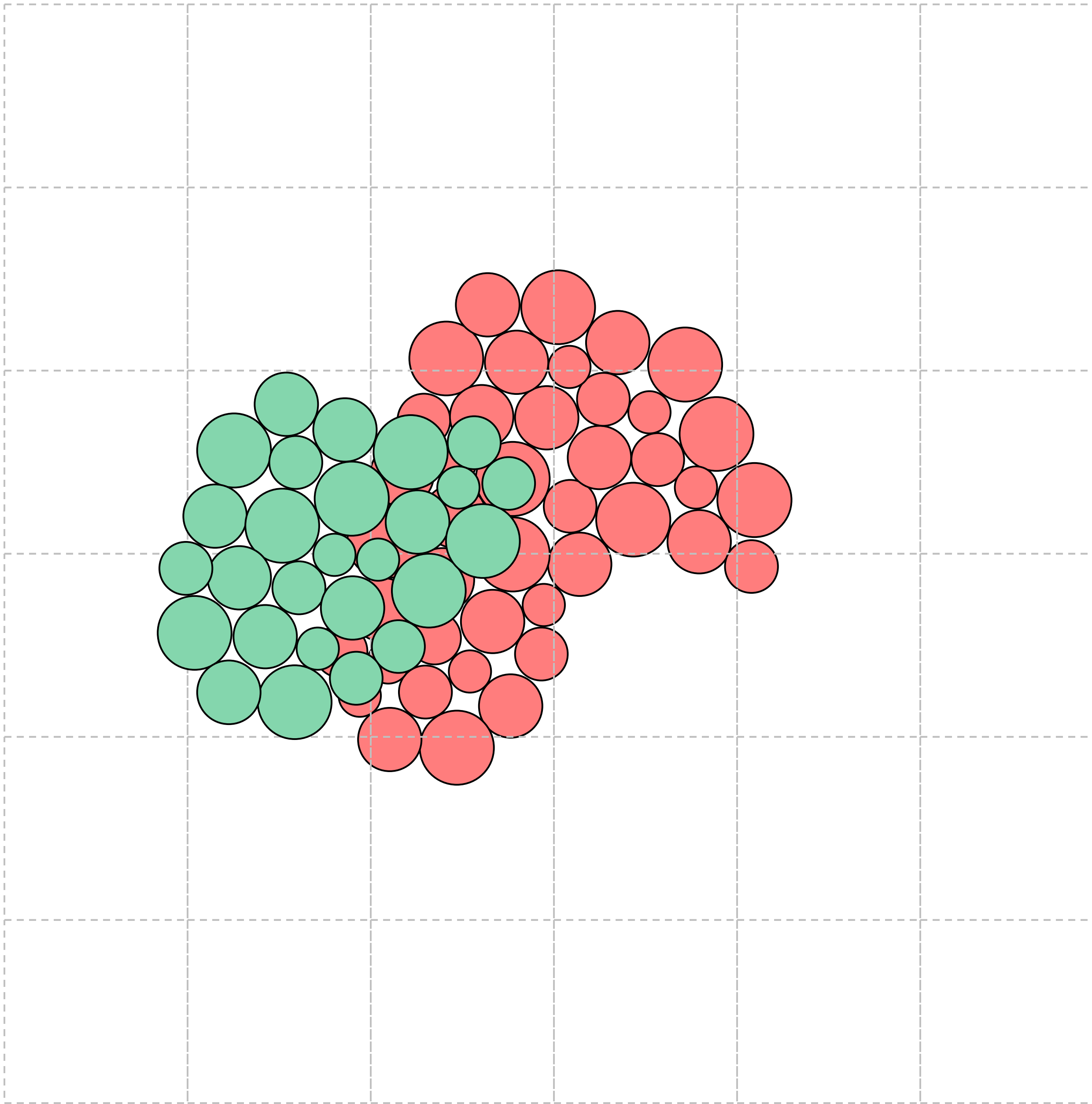
Docking: Rotational & Translational Search



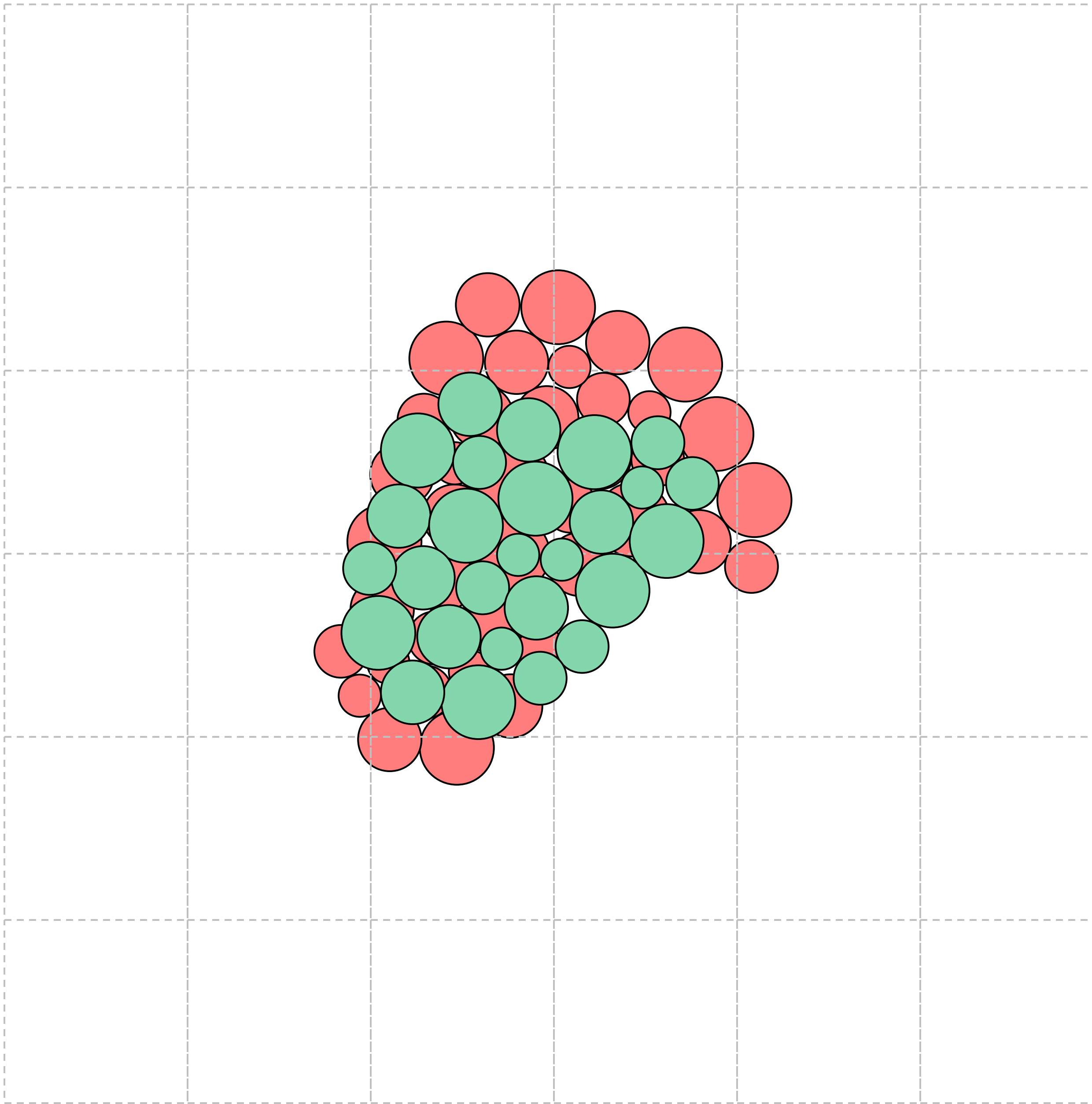
Docking: Rotational & Translational Search



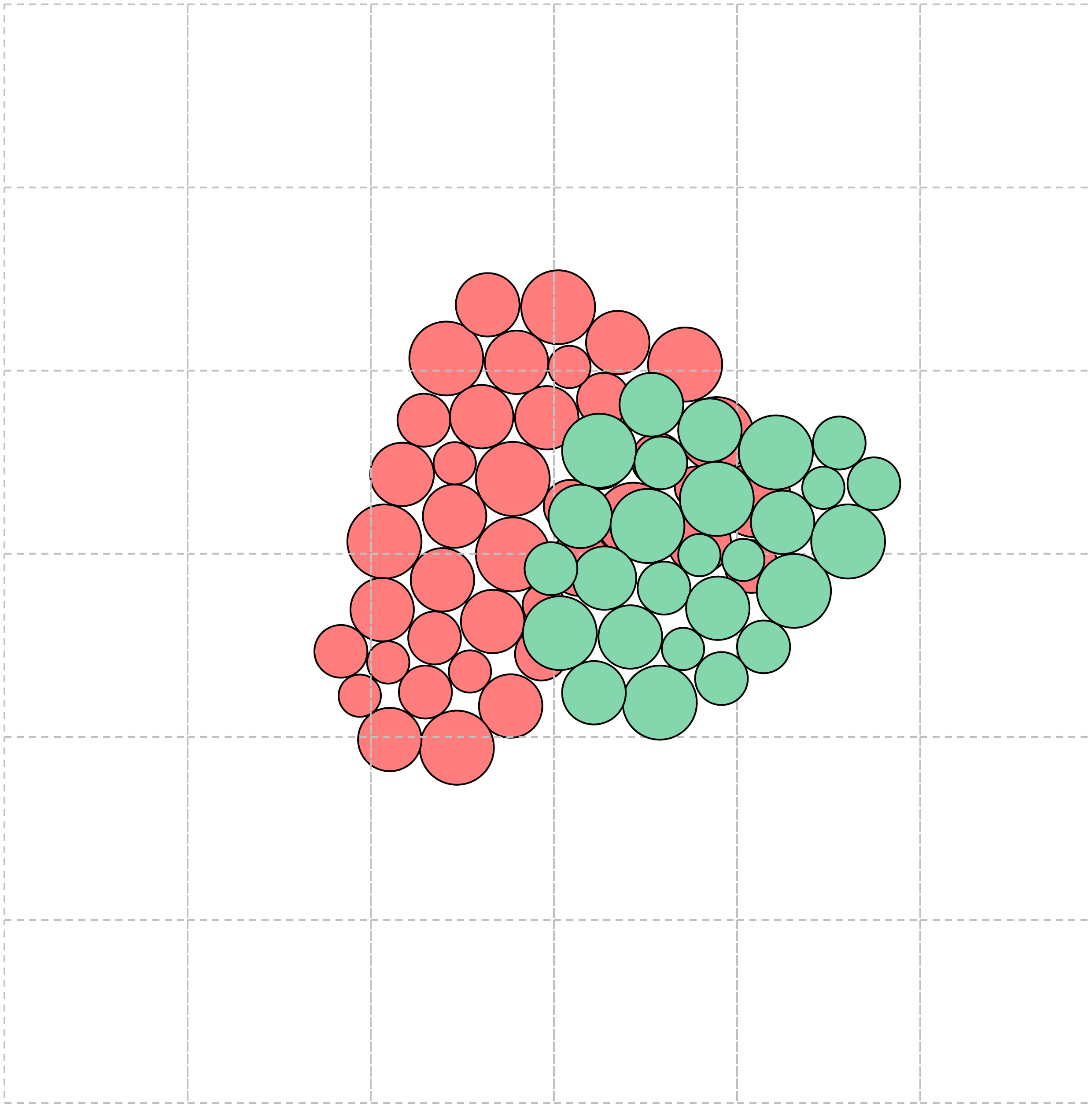
Docking: Rotational & Translational Search



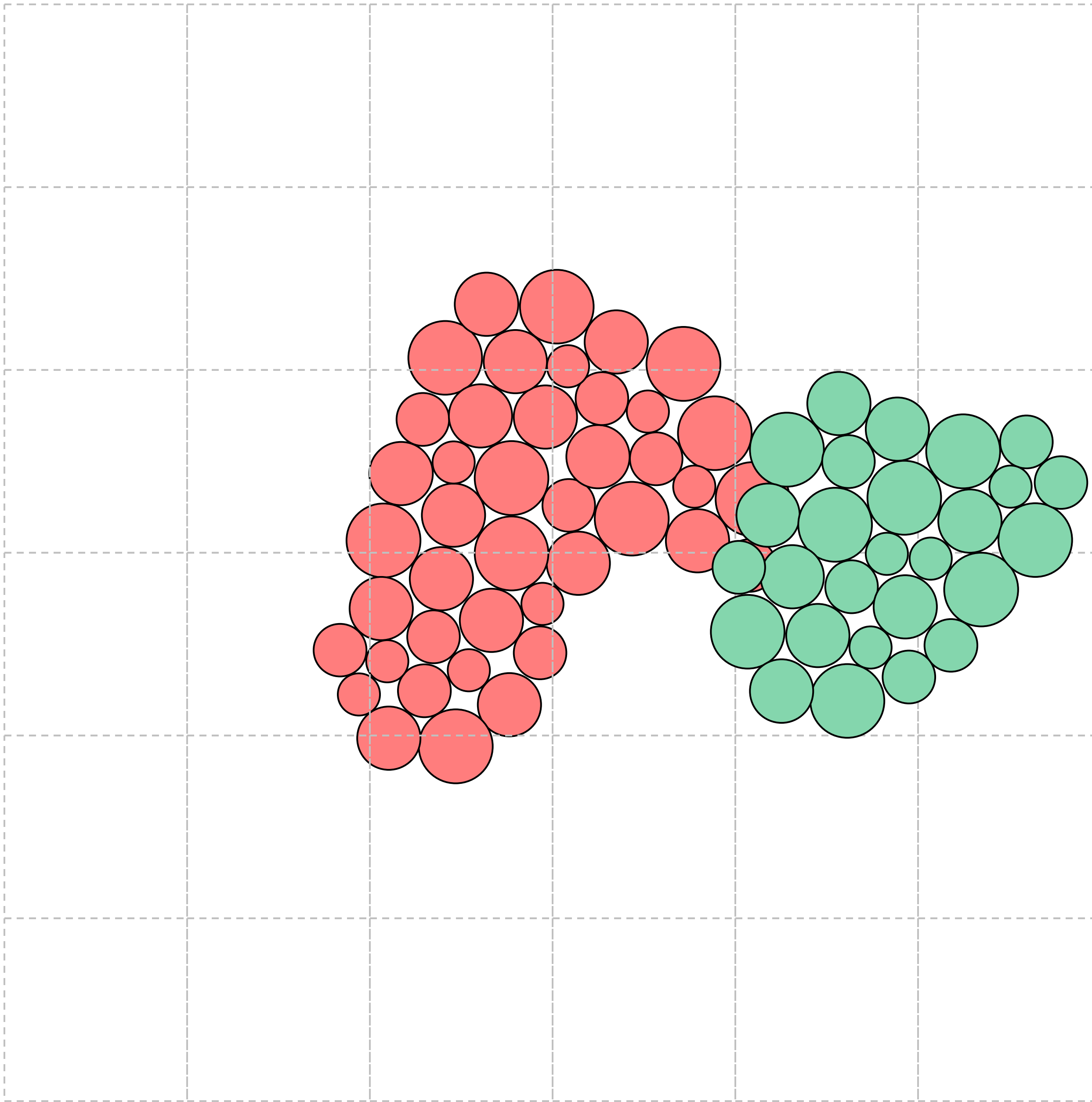
Docking: Rotational & Translational Search



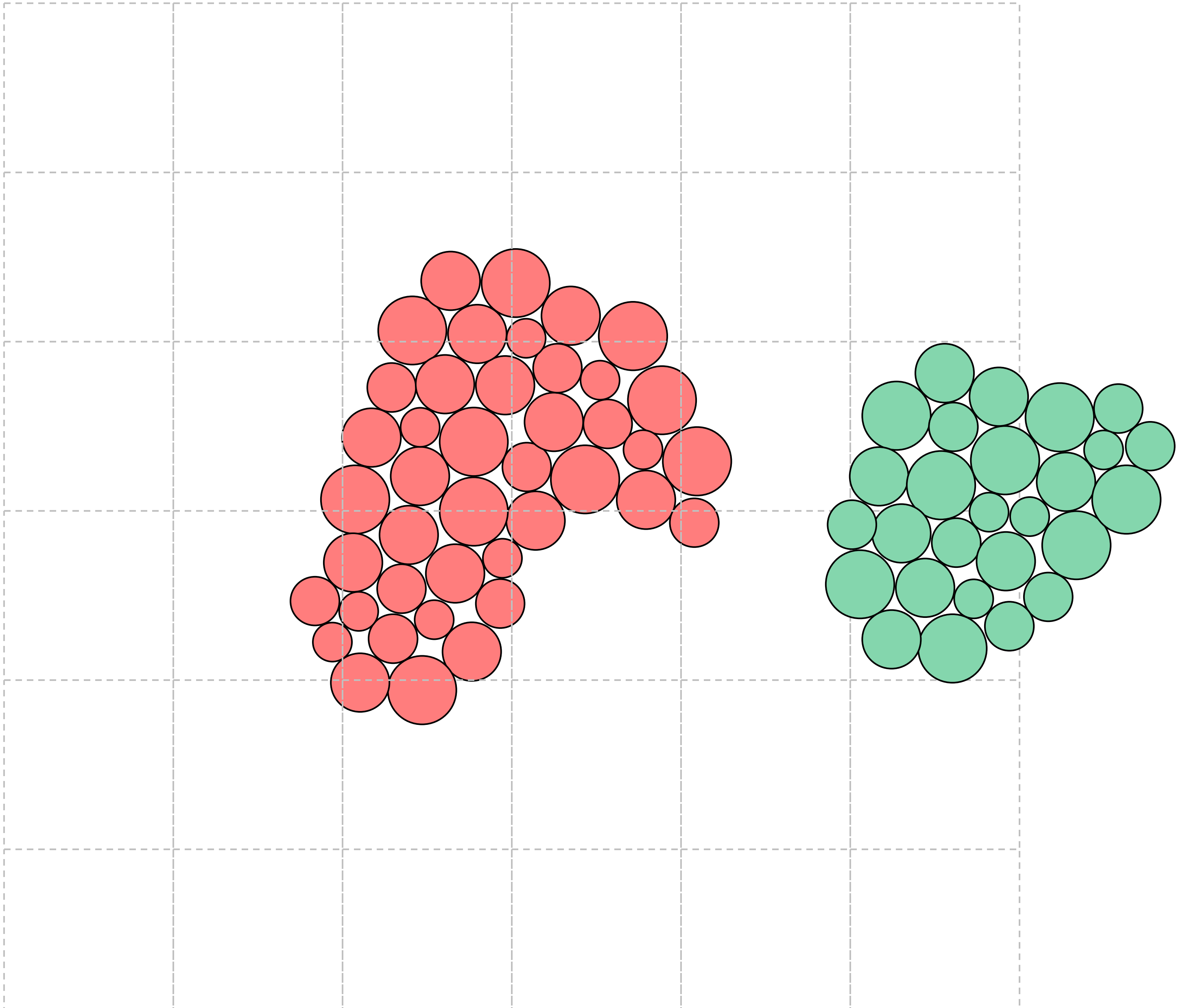
Docking: Rotational & Translational Search



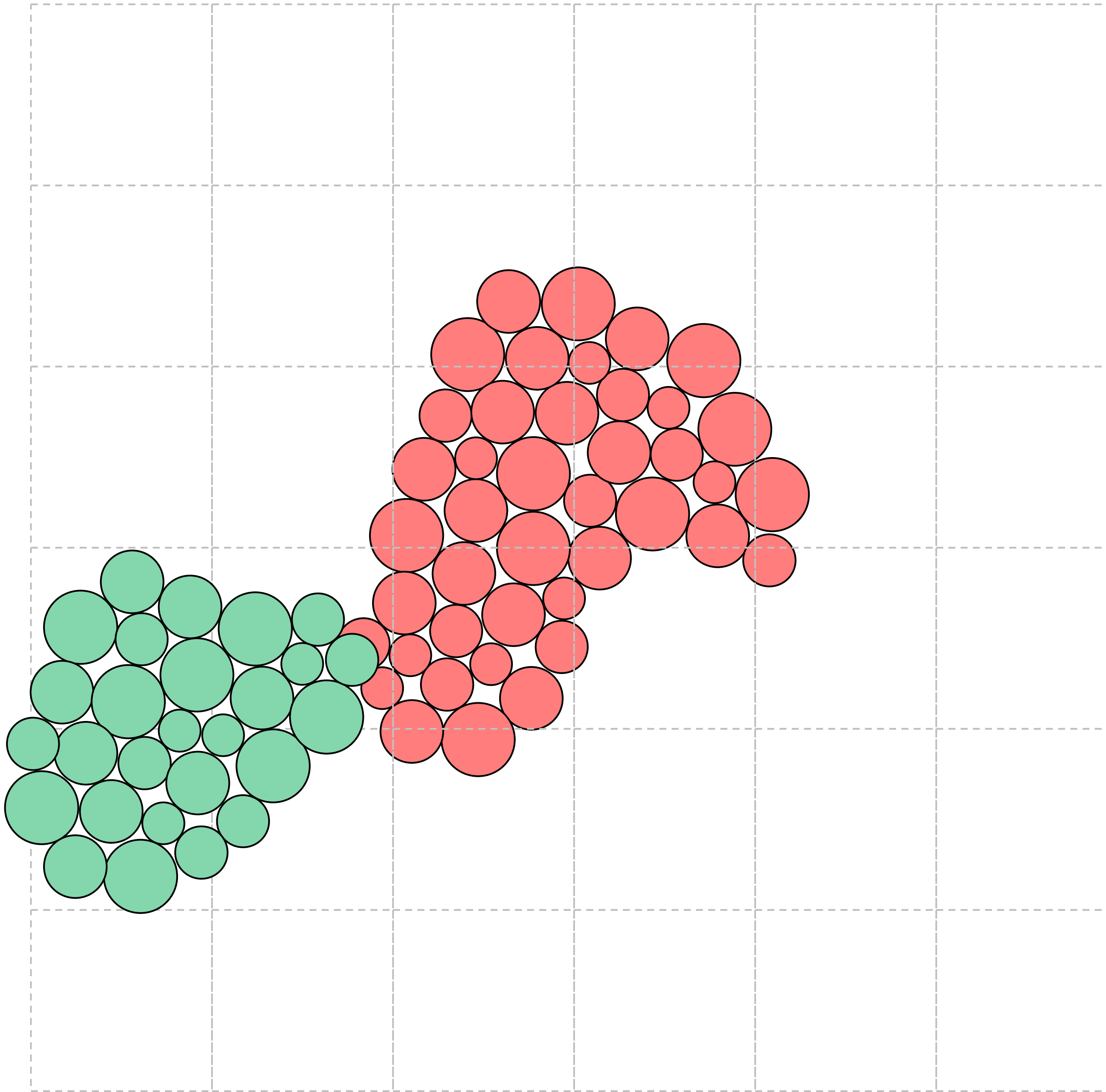
Docking: Rotational & Translational Search



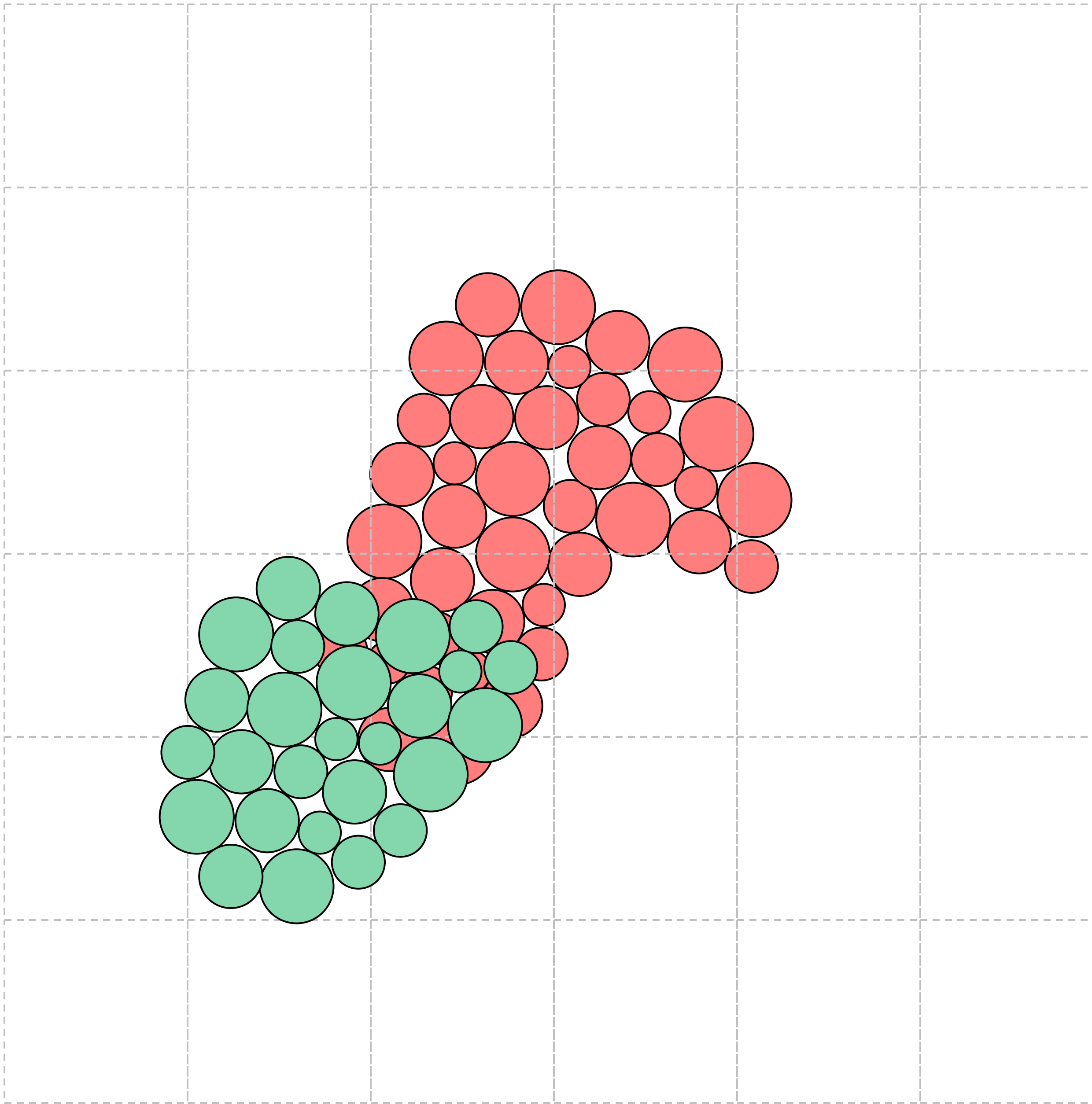
Docking: Rotational & Translational Search



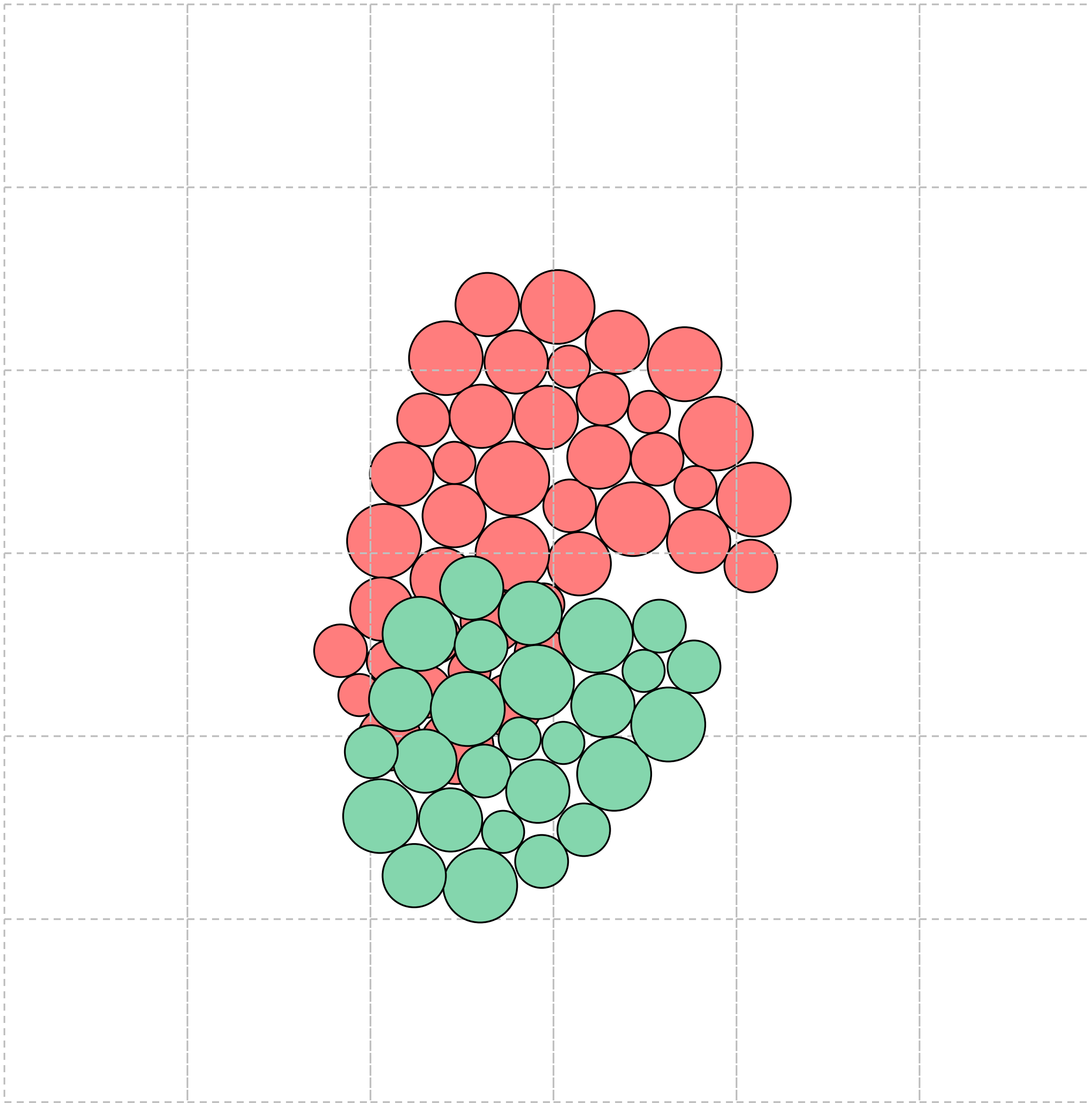
Docking: Rotational & Translational Search



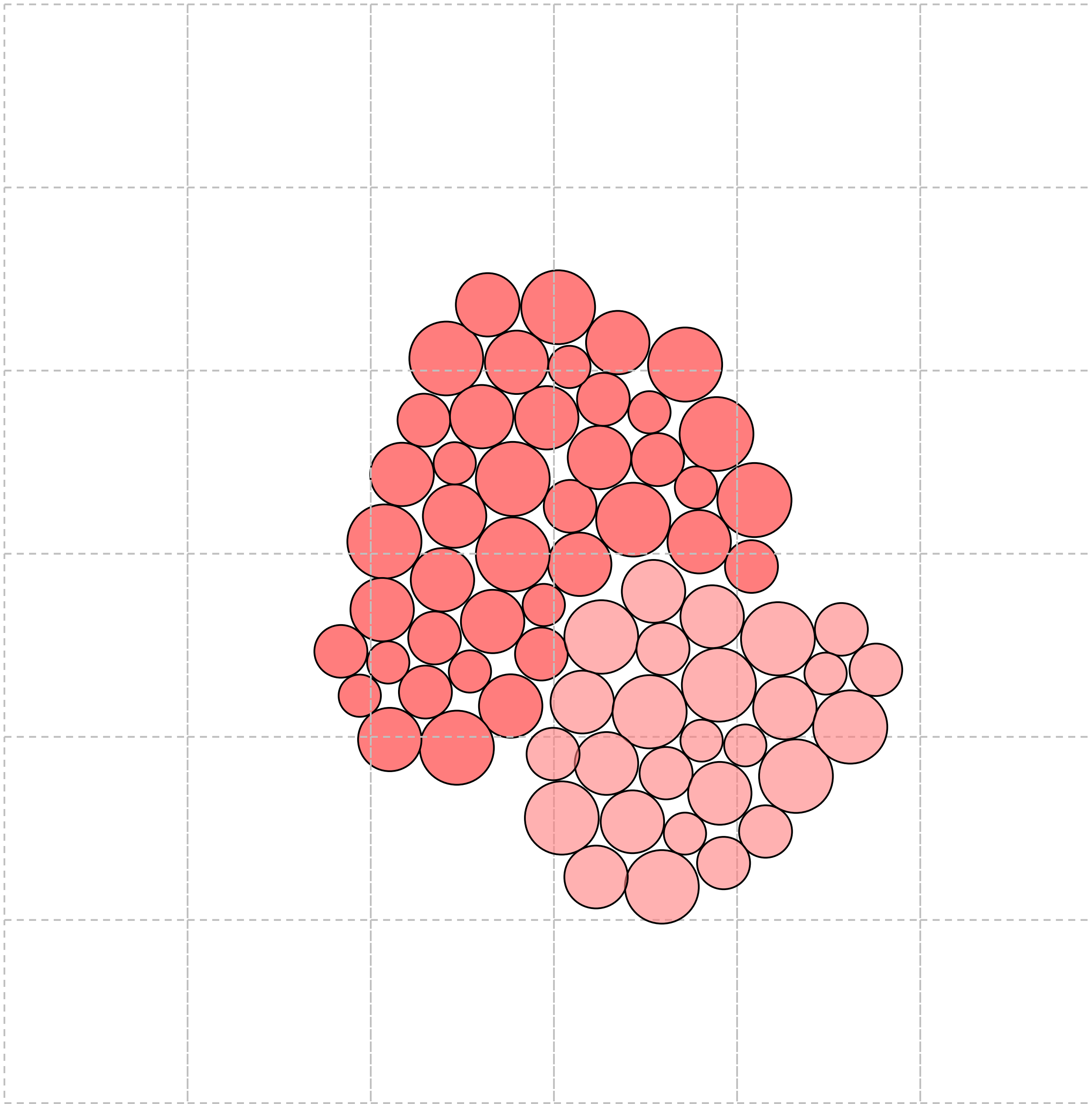
Docking: Rotational & Translational Search



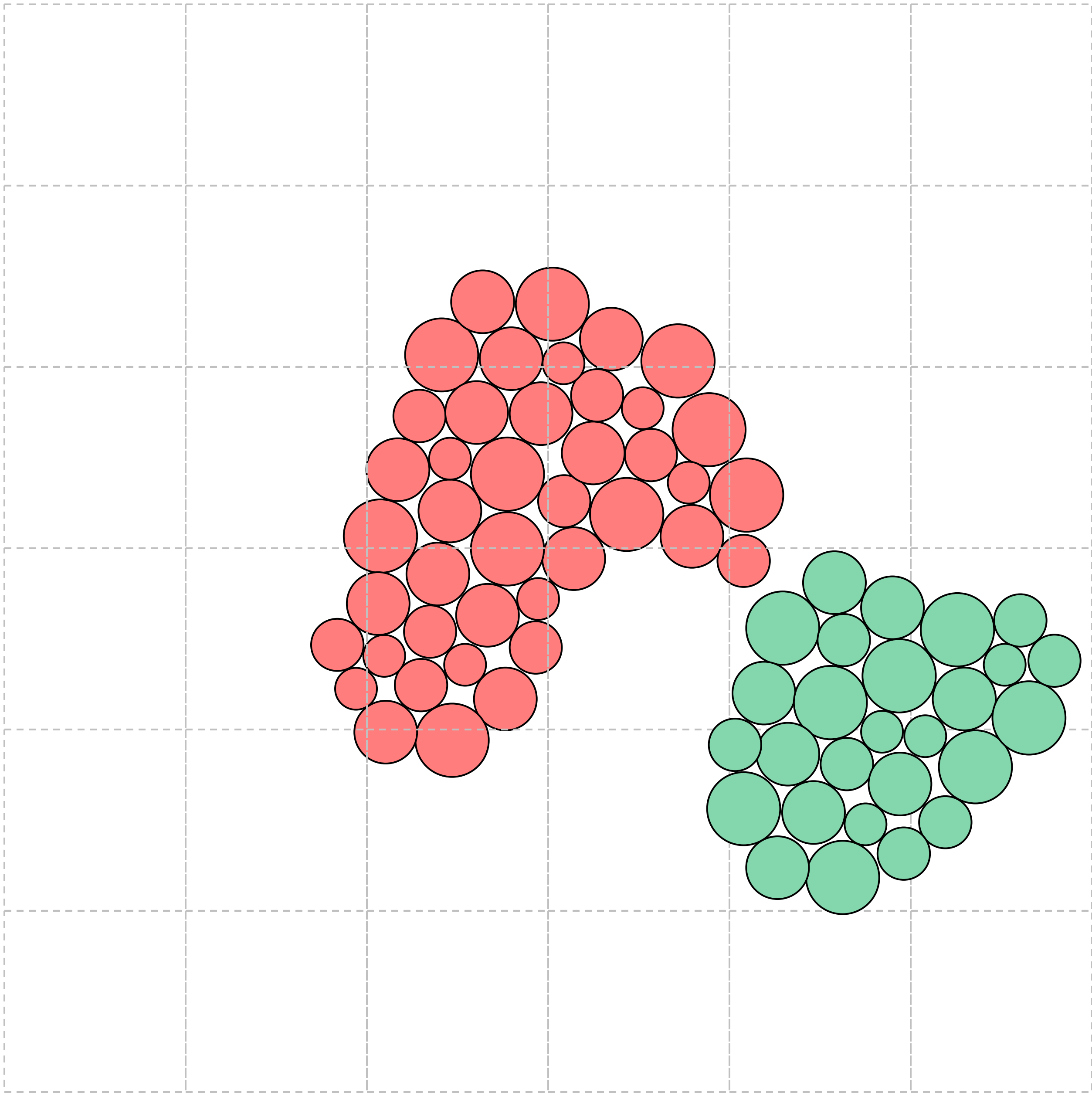
Docking: Rotational & Translational Search



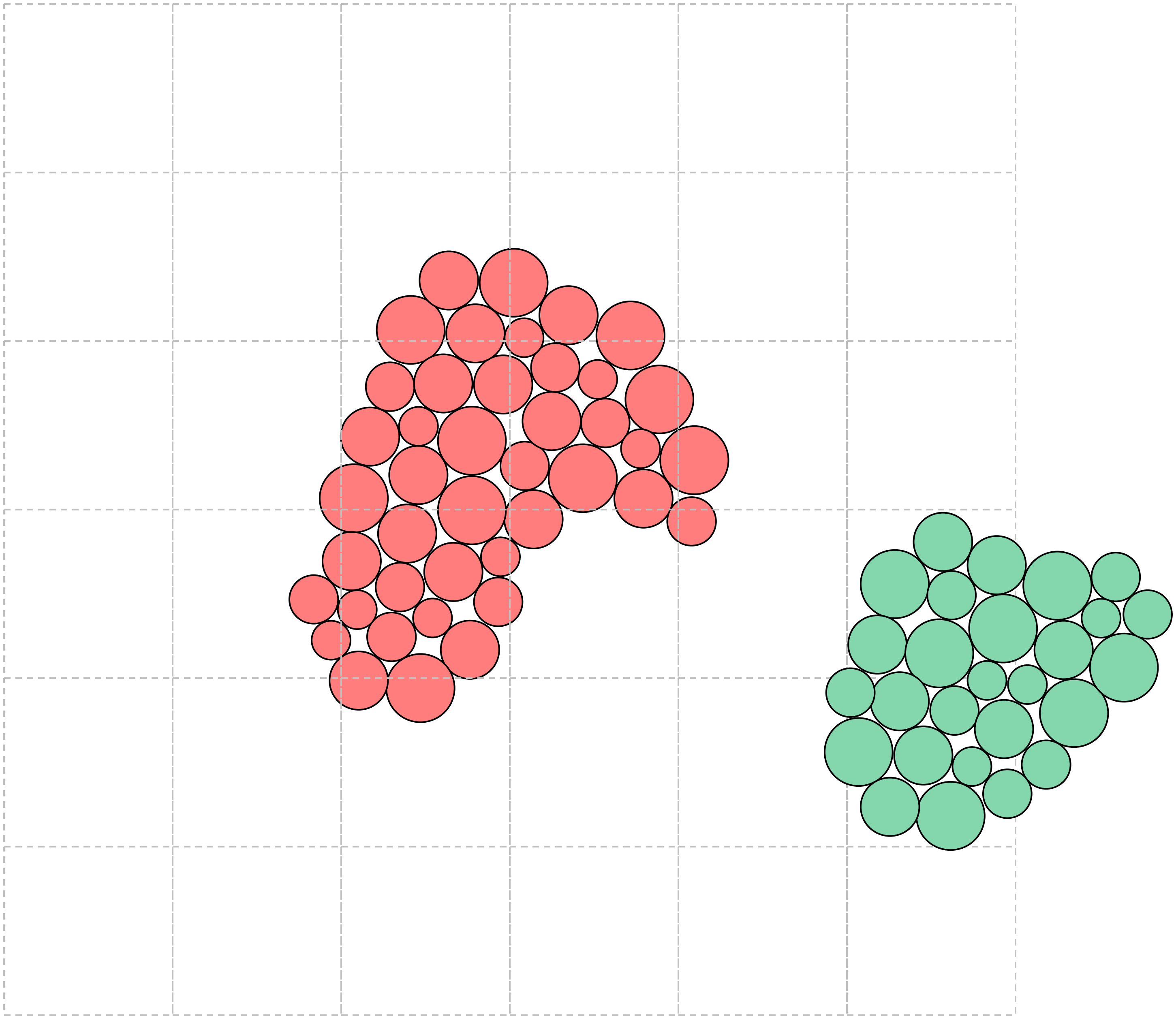
Docking: Rotational & Translational Search



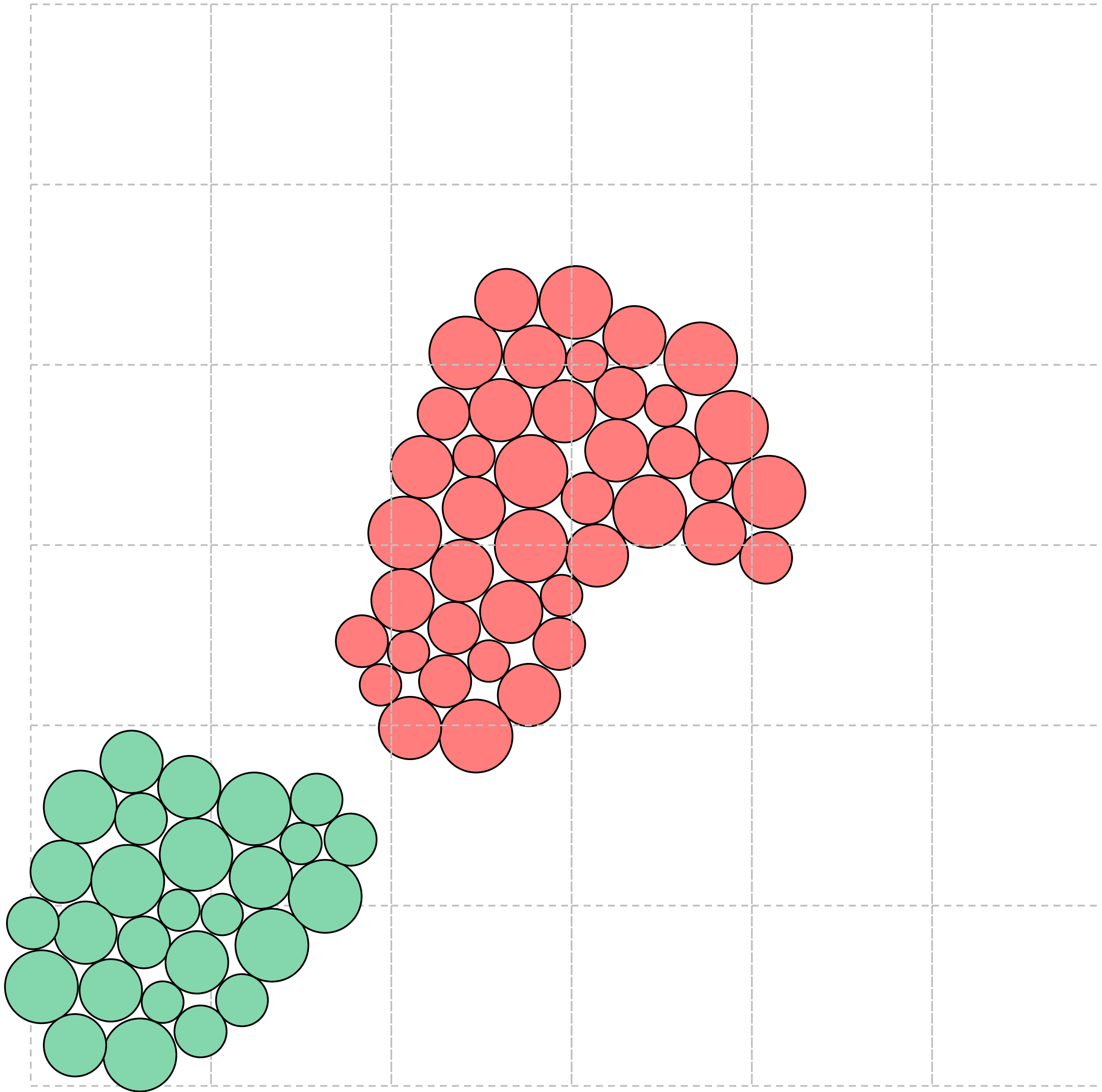
Docking: Rotational & Translational Search



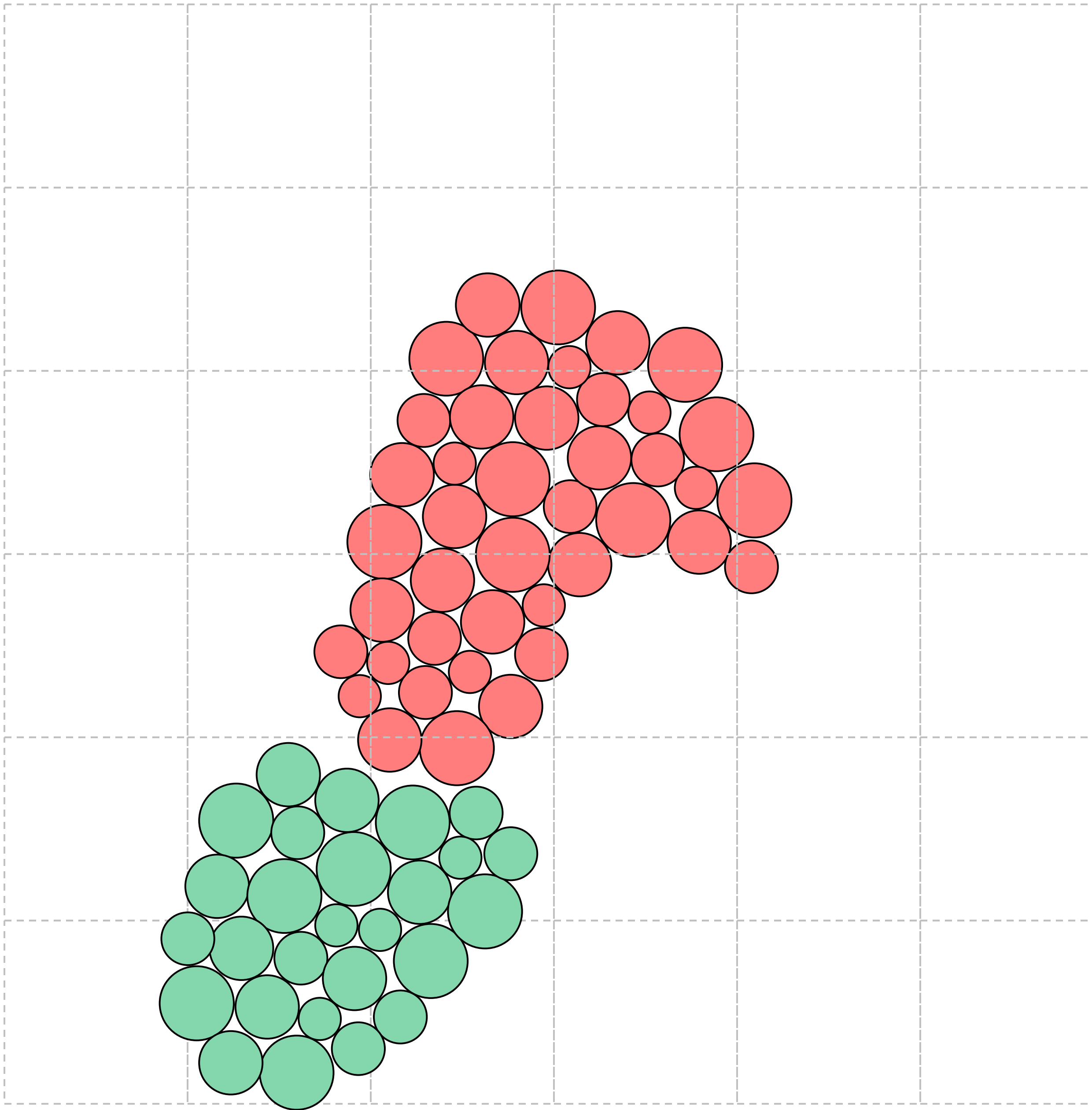
Docking: Rotational & Translational Search



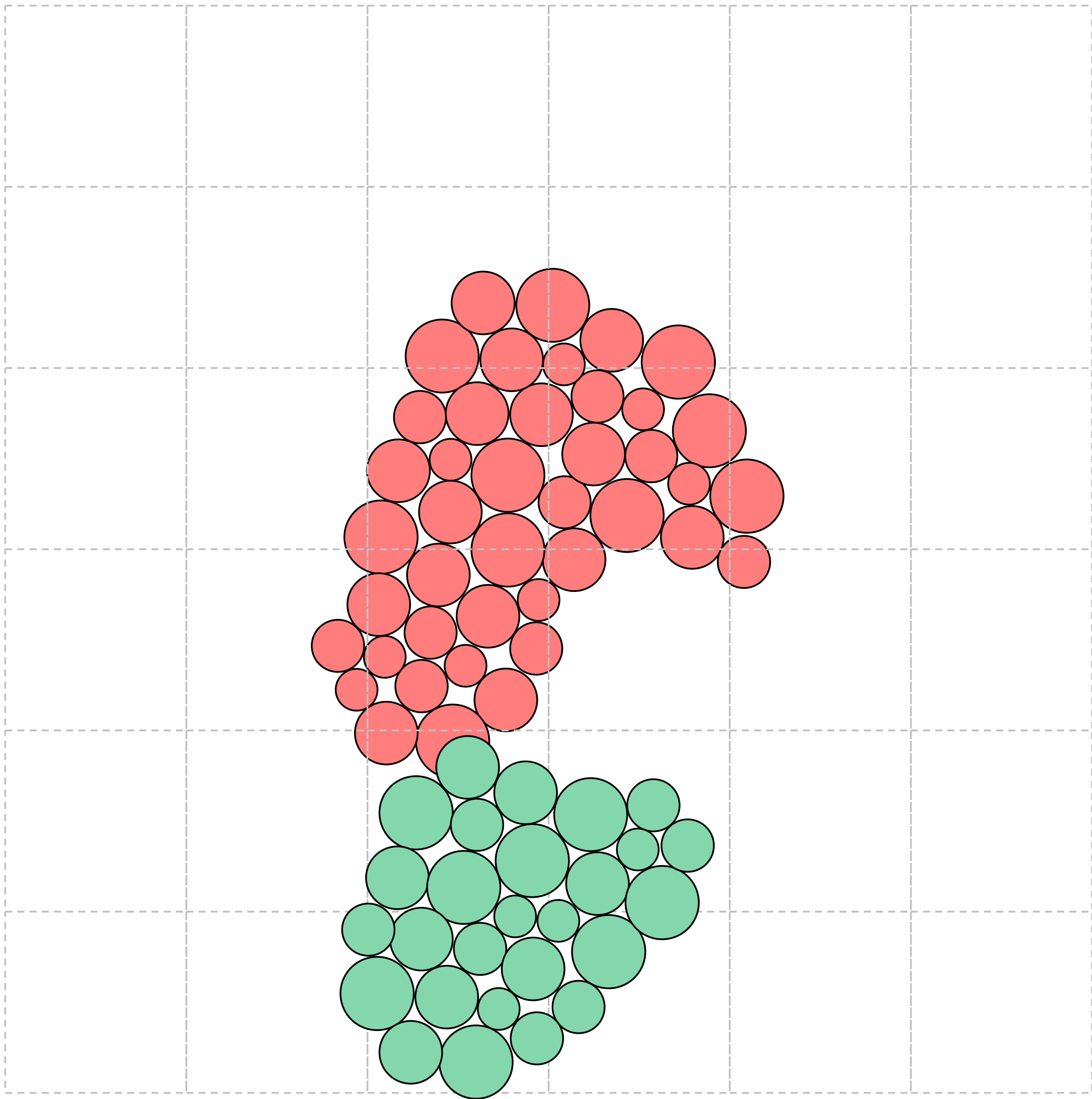
Docking: Rotational & Translational Search



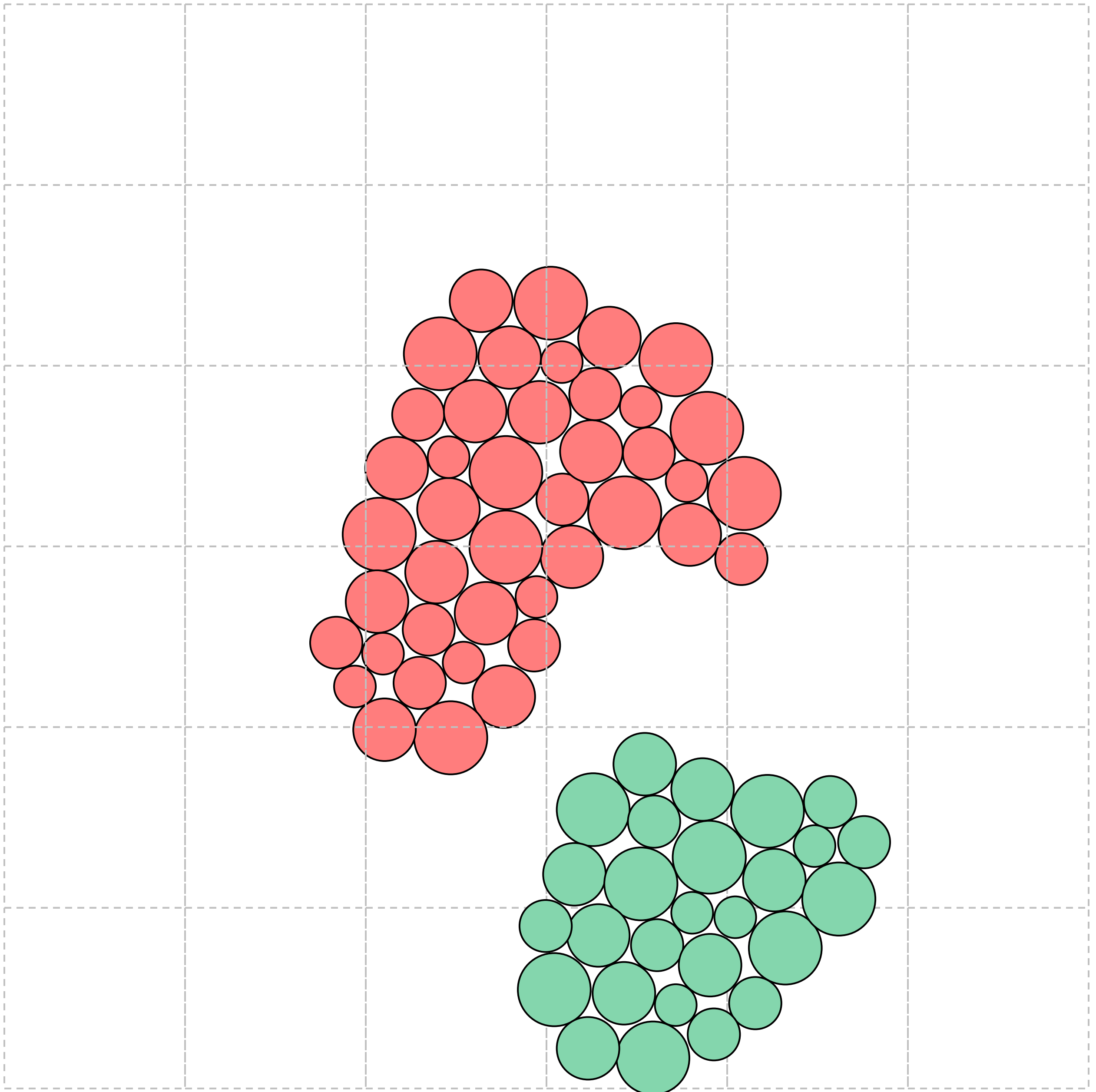
Docking: Rotational & Translational Search



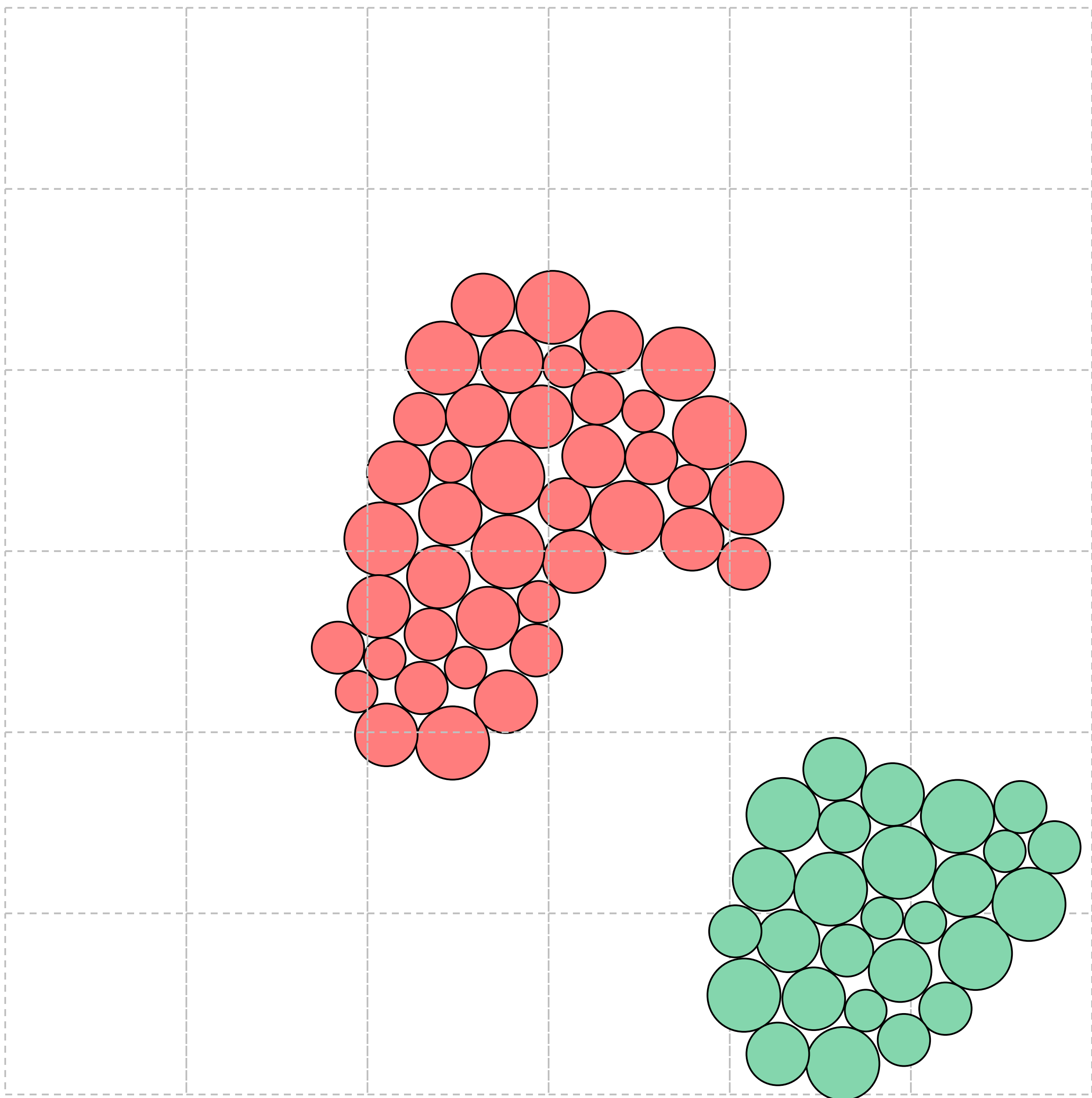
Docking: Rotational & Translational Search



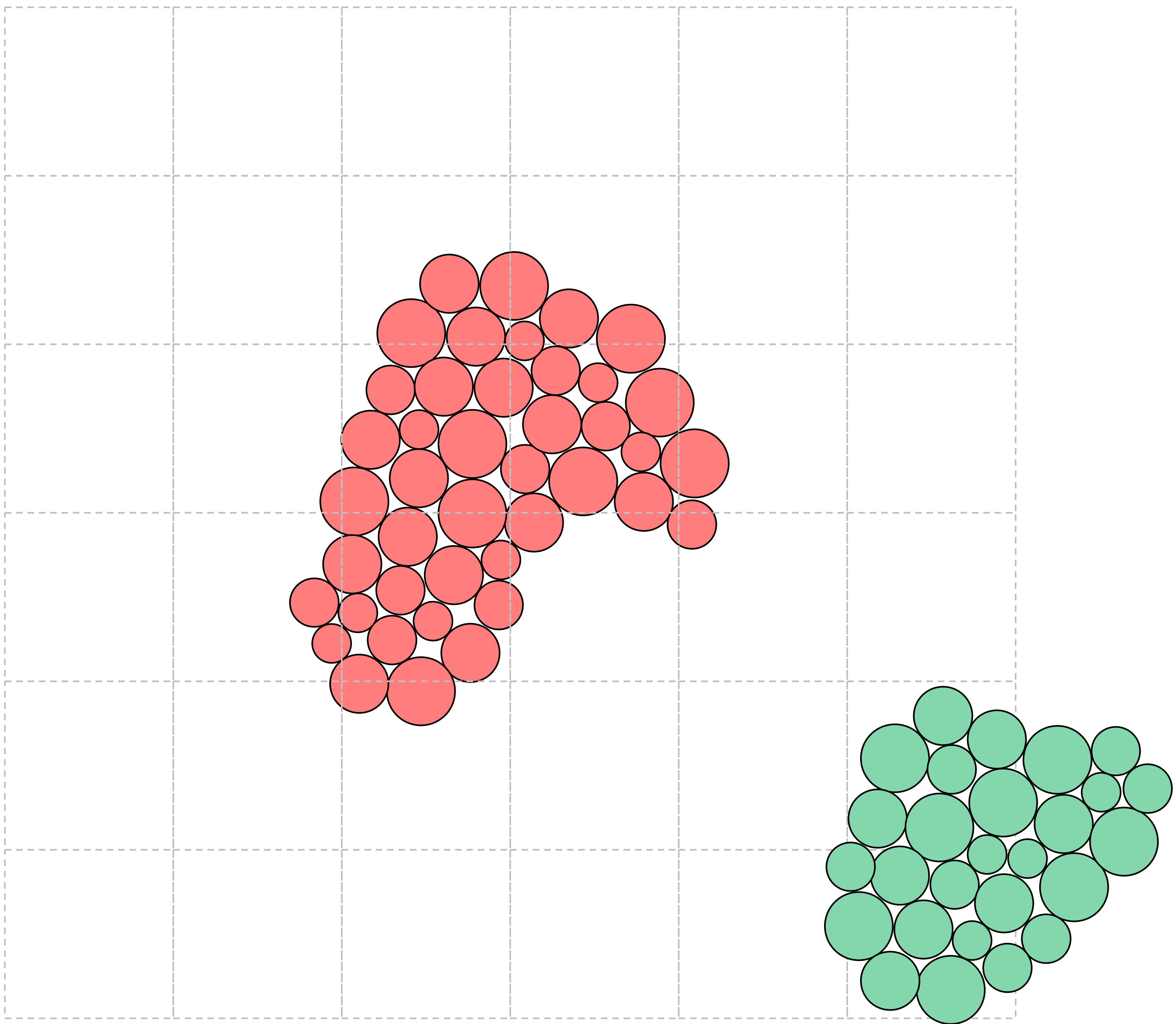
Docking: Rotational & Translational Search



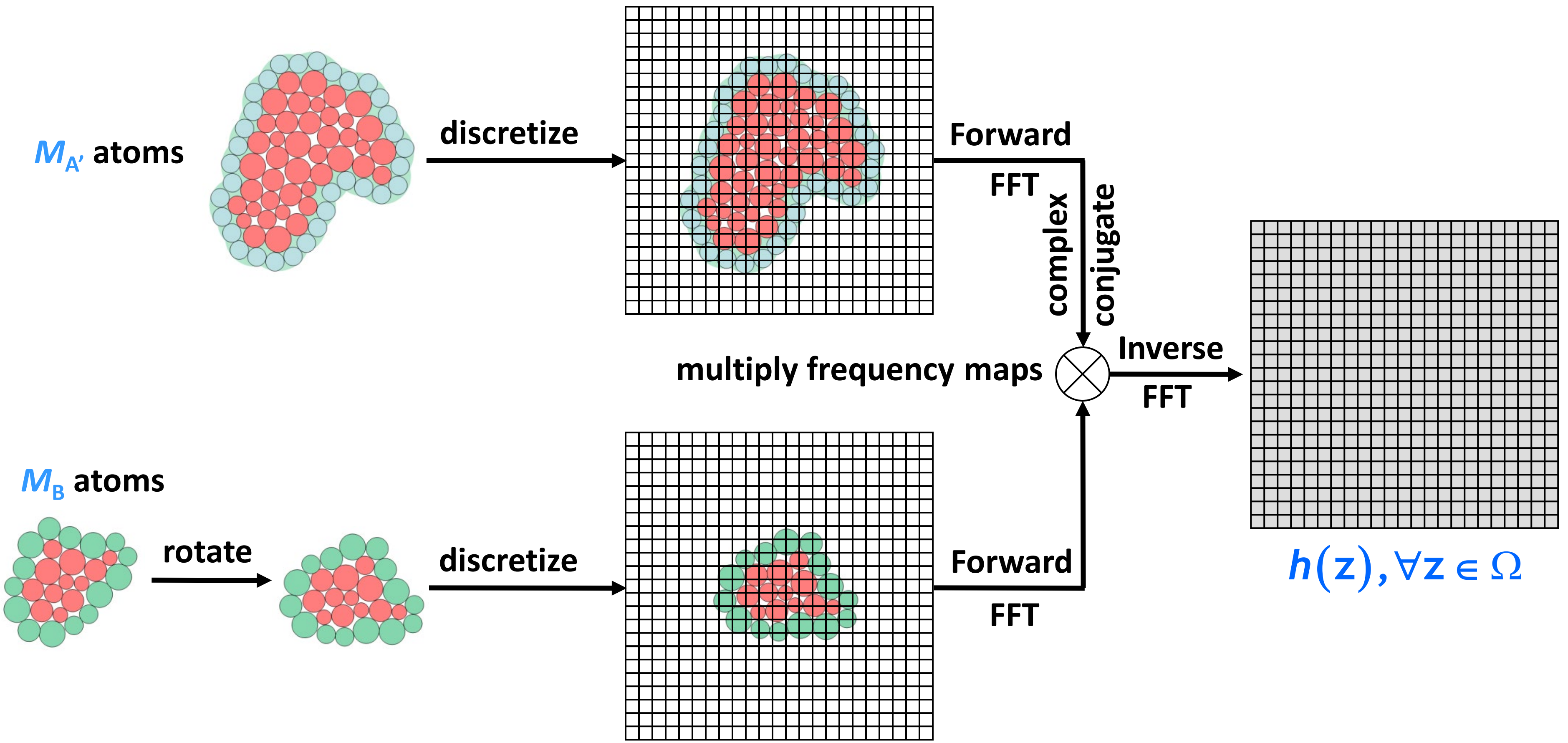
Docking: Rotational & Translational Search



Docking: Rotational & Translational Search



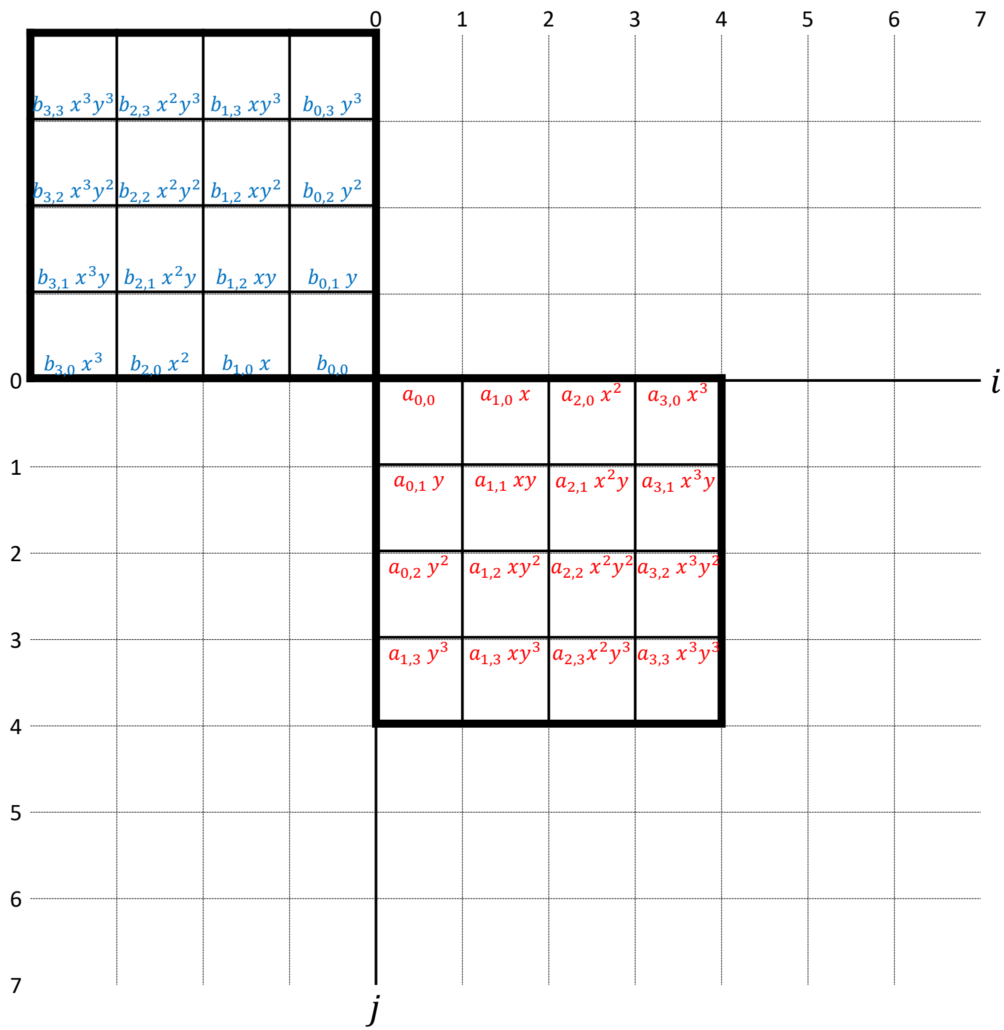
Translational Search using FFT



$$\forall z \in \Omega = [-n, n]^3, \qquad h(z) = \int_{x \in \Omega} f_{A'}(x) f_{B_r}(z - x) dx$$

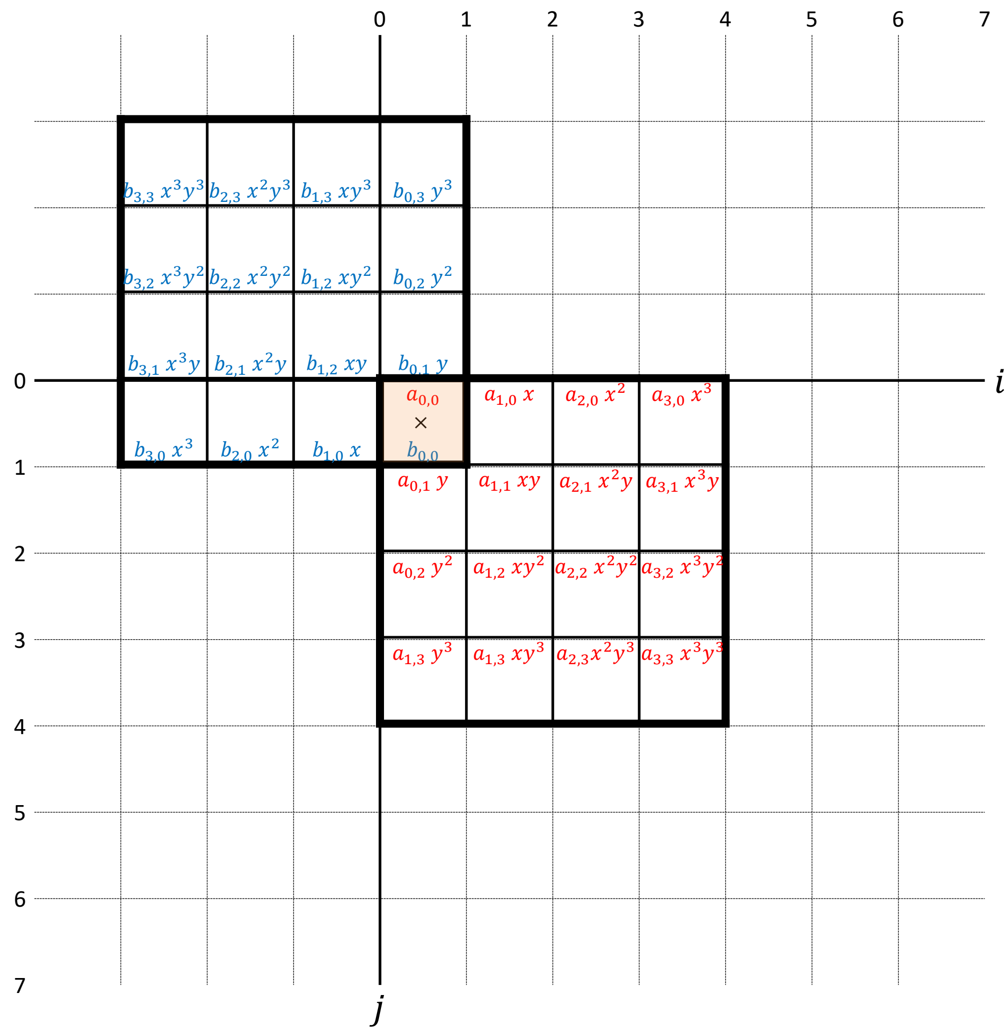
Multiplication of Bivariate Polynomials

Translation $\langle i, j \rangle$



Coefficient of $x^{i-1}y^{j-1}$

Multiplication of Bivariate Polynomials



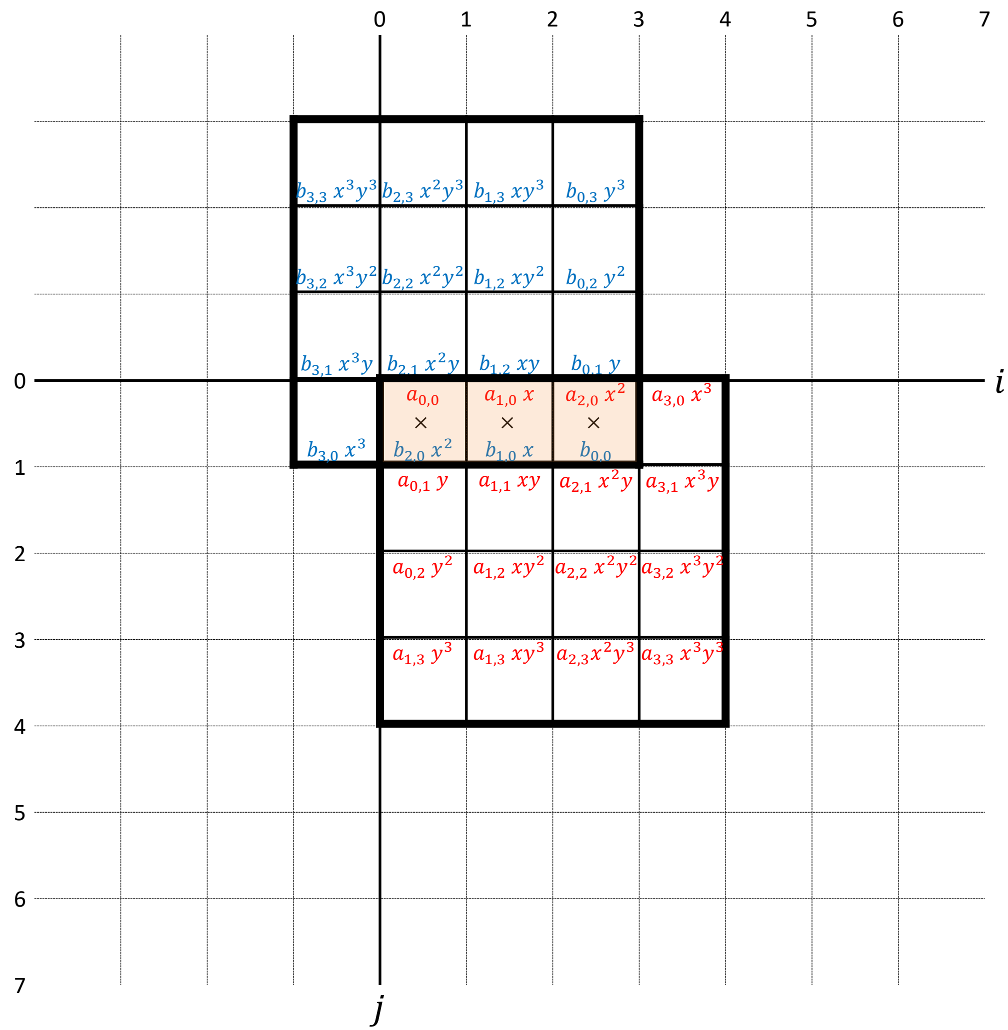
Translation $\langle i,j\rangle$

Translation $\langle 1,1\rangle$

Coefficient of $x^{i-1}y^{j-1}$

Coefficient of x^0y^0

Multiplication of Bivariate Polynomials



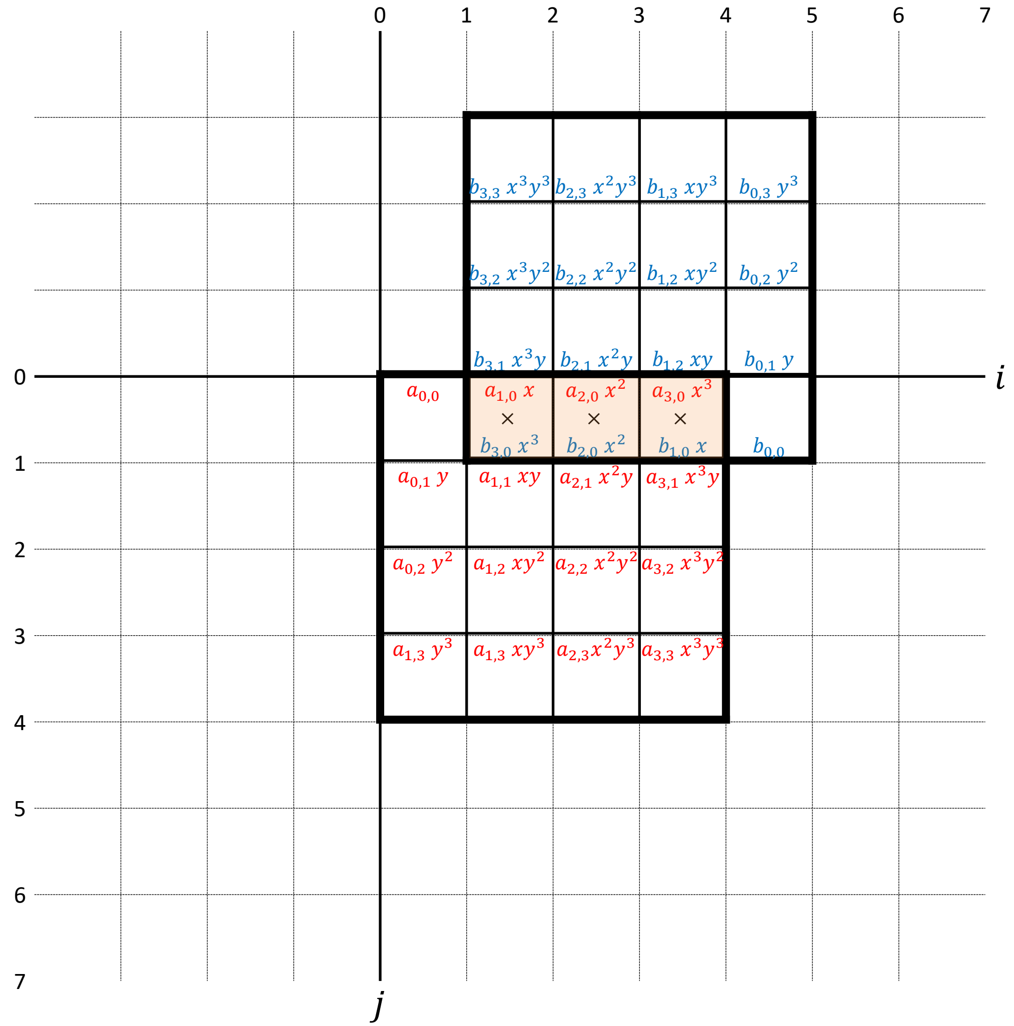
Translation $\langle i, j\rangle$

Translation $\langle 3,1\rangle$

Coefficient of $x^{i-1}y^{j-1}$

Coefficient of x^2y^0

Multiplication of Bivariate Polynomials



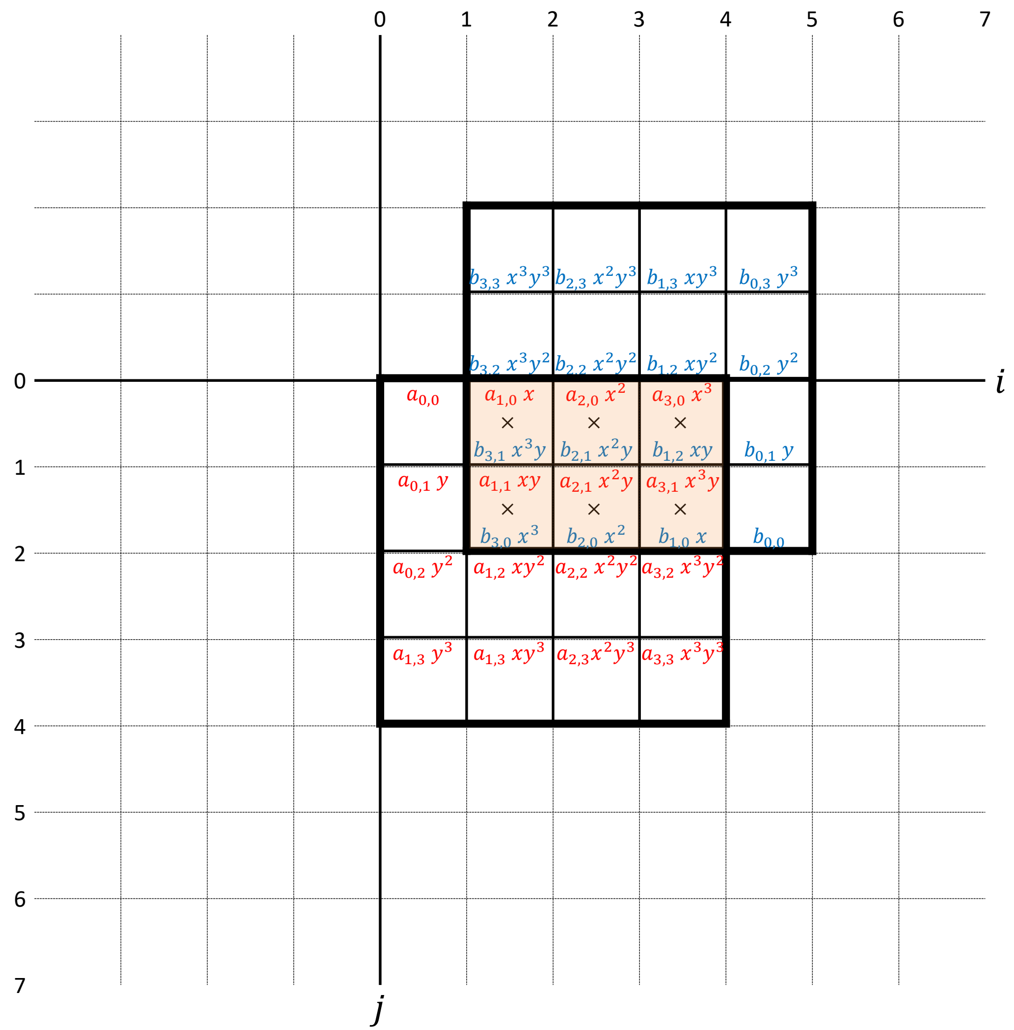
Translation $\langle i, j\rangle$

Translation $\langle 5,1\rangle$

Coefficient of $x^{i-1}y^{j-1}$

Coefficient of x^4y^0

Multiplication of Bivariate Polynomials



Translation $\langle i, j\rangle$

Translation $\langle 5,2\rangle$

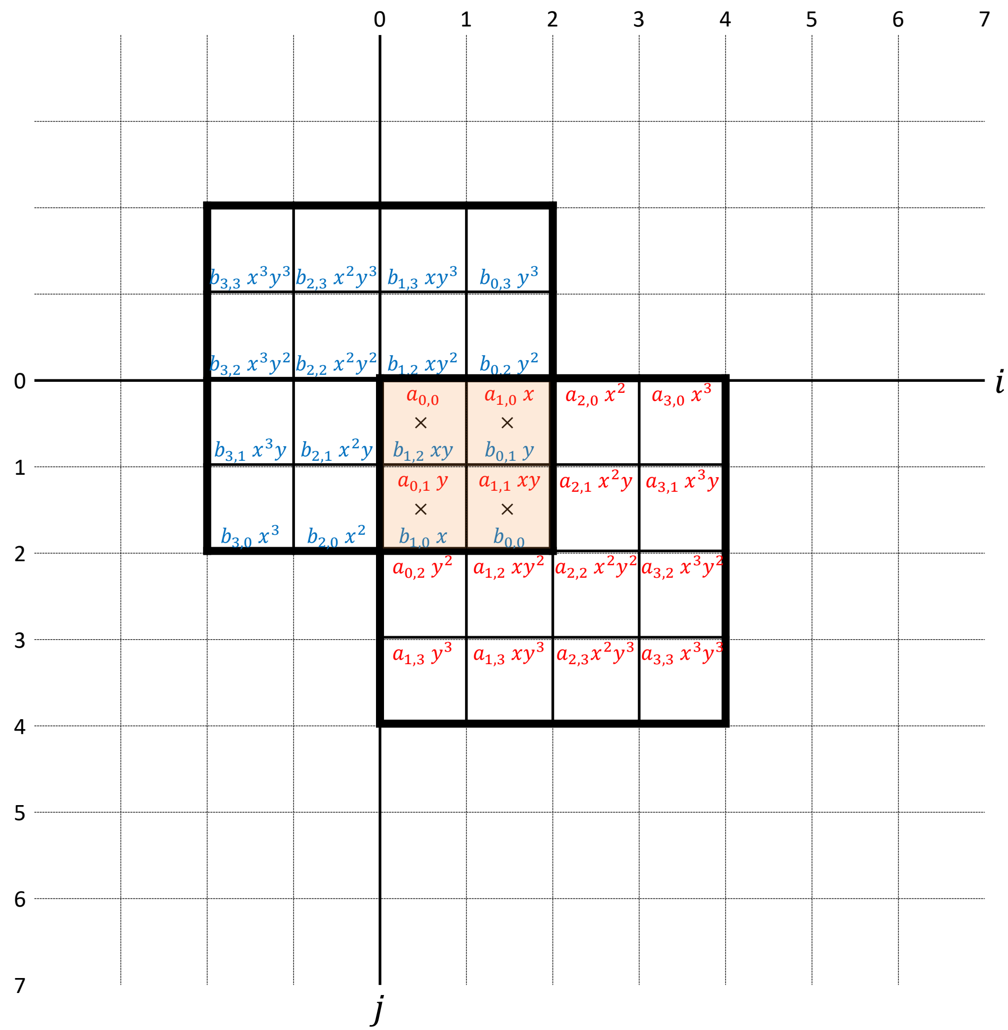
Coefficient of $x^{i-1}y^{j-1}$

Coefficient of x^4y^1

Multiplication of Bivariate Polynomials

Translation $\langle i, j \rangle$

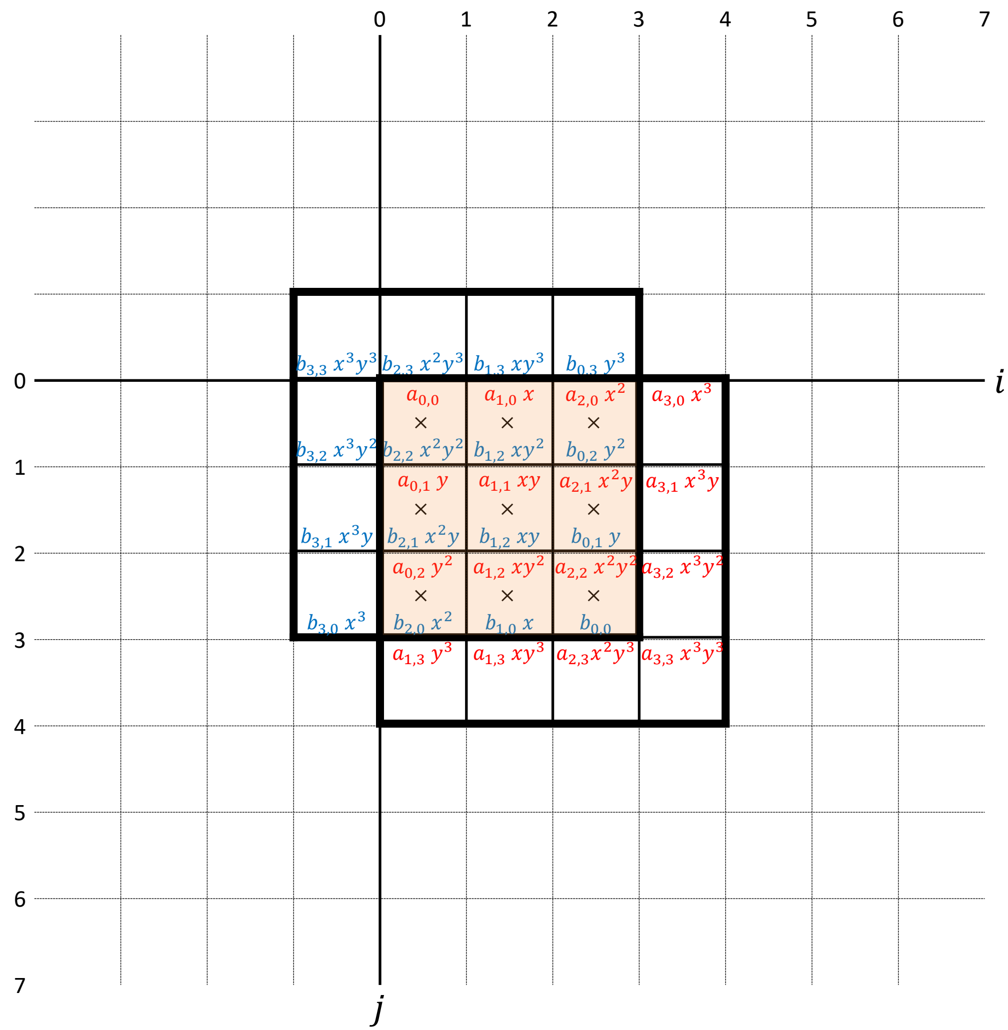
Translation $\langle 2,2 \rangle$



Coefficient of $x^{i-1}y^{j-1}$

Coefficient of x^1y^1

Multiplication of Bivariate Polynomials



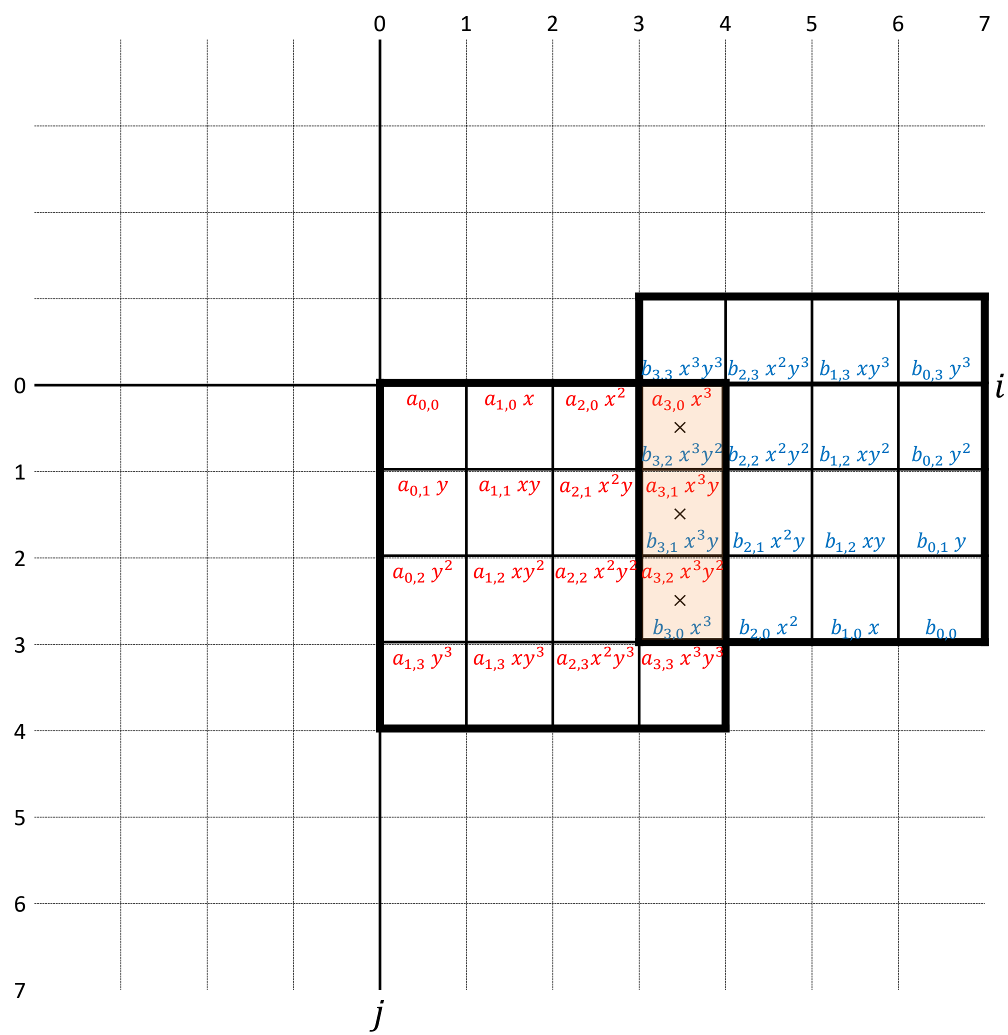
Translation $\langle i,j\rangle$

Translation $\langle 3,3\rangle$

Coefficient of $x^{i-1}y^{j-1}$

Coefficient of x^2y^2

Multiplication of Bivariate Polynomials



Translation $\langle i, j\rangle$

Translation $\langle 7,3\rangle$

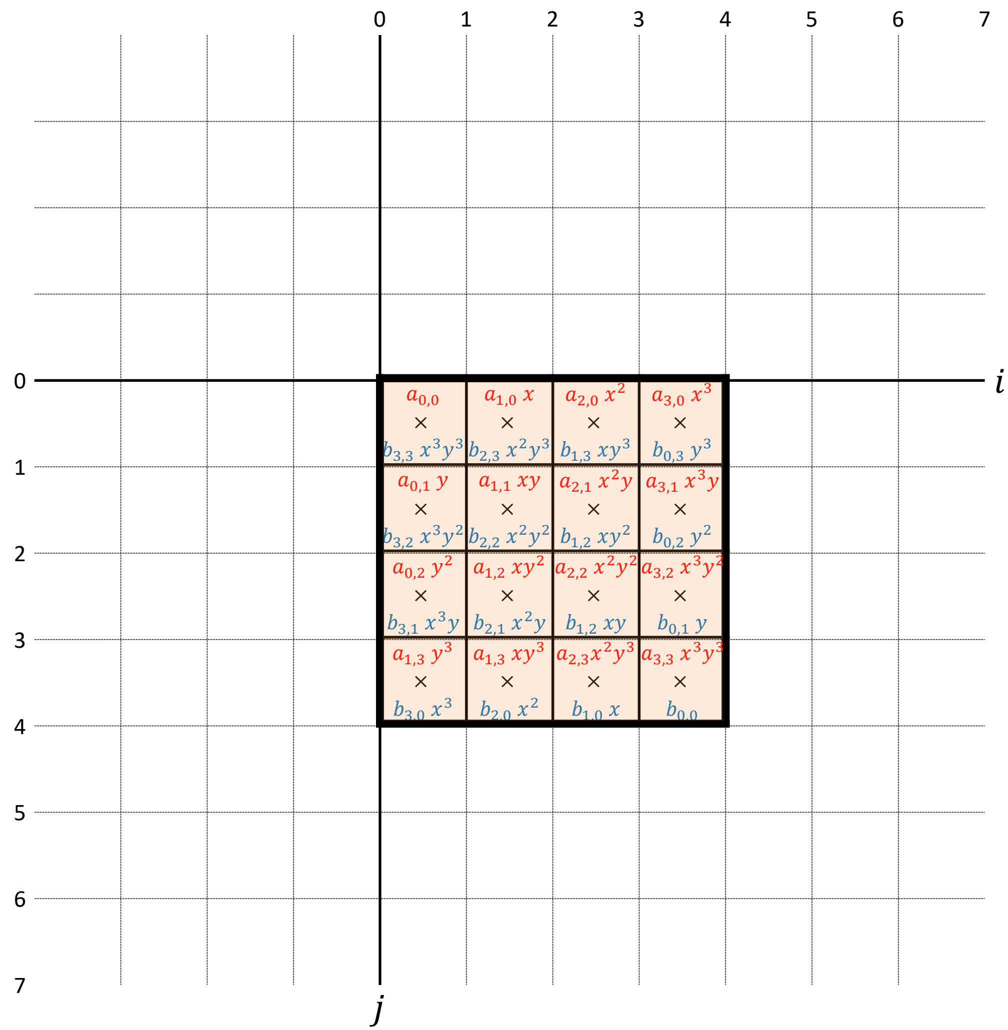
Coefficient of $x^{i-1}y^{j-1}$

Coefficient of x^6y^2

Multiplication of Bivariate Polynomials

Translation $\langle i, j \rangle$

Translation $\langle 4,4 \rangle$



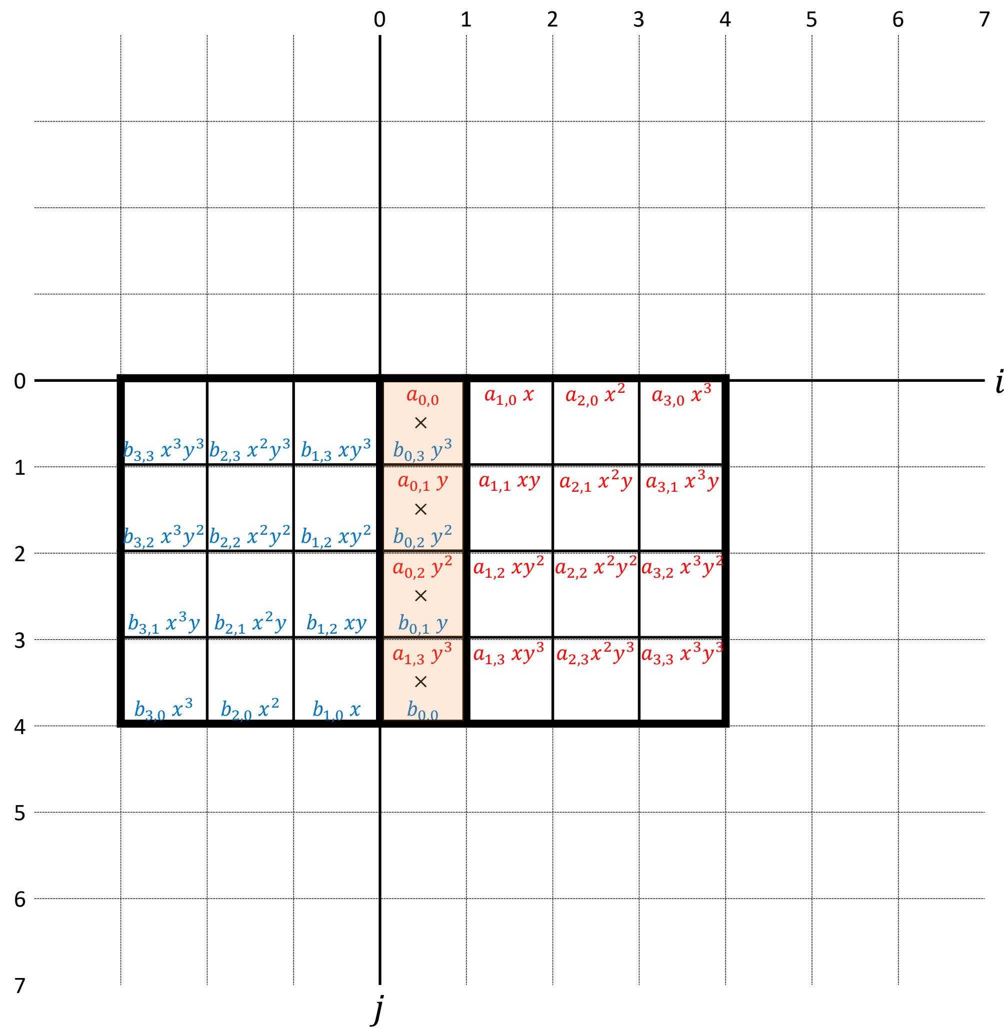
Coefficient of $x^{i-1}y^{j-1}$

Coefficient of x^3y^3

Multiplication of Bivariate Polynomials

Translation $\langle i, j \rangle$

Translation $\langle 1,4 \rangle$



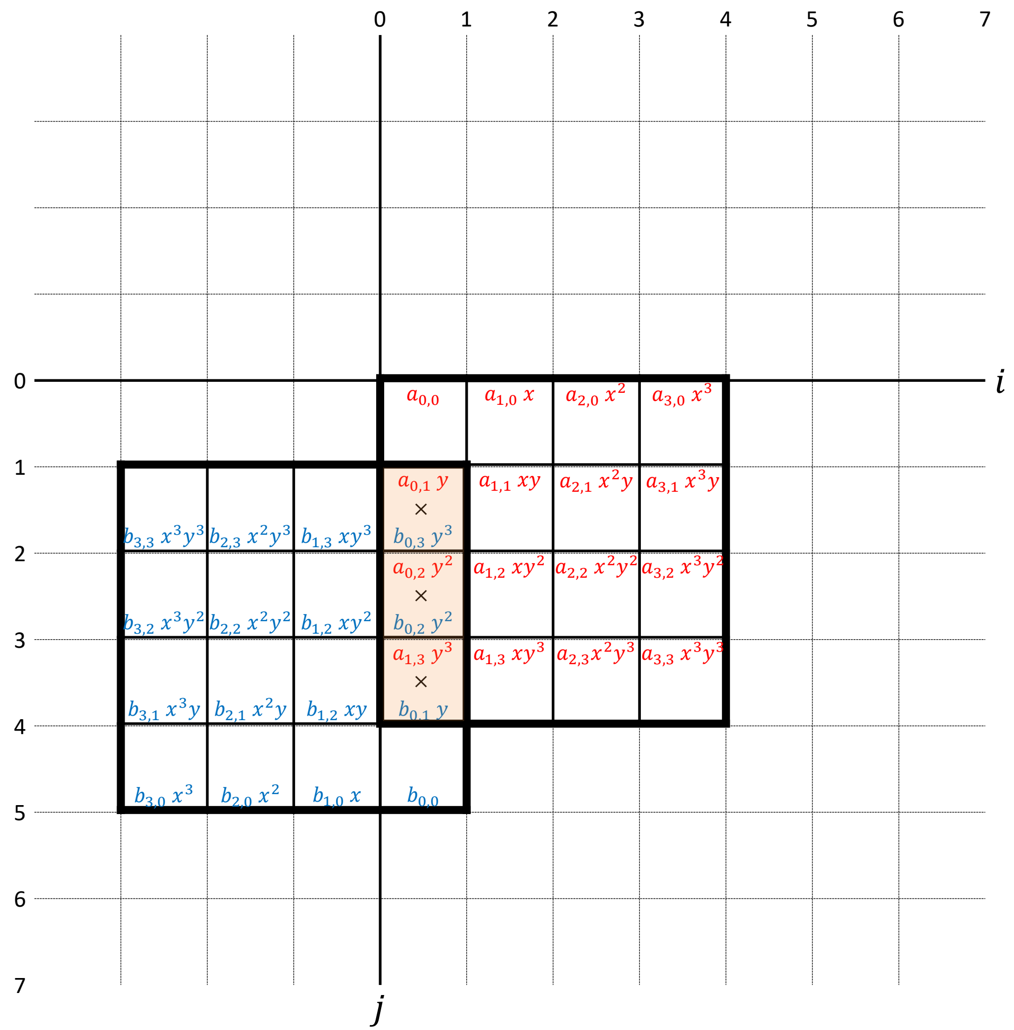
Coefficient of $x^{i-1}y^{j-1}$

Coefficient of x^0y^3

Multiplication of Bivariate Polynomials

Translation $\langle i, j \rangle$

Translation $\langle 1,5 \rangle$



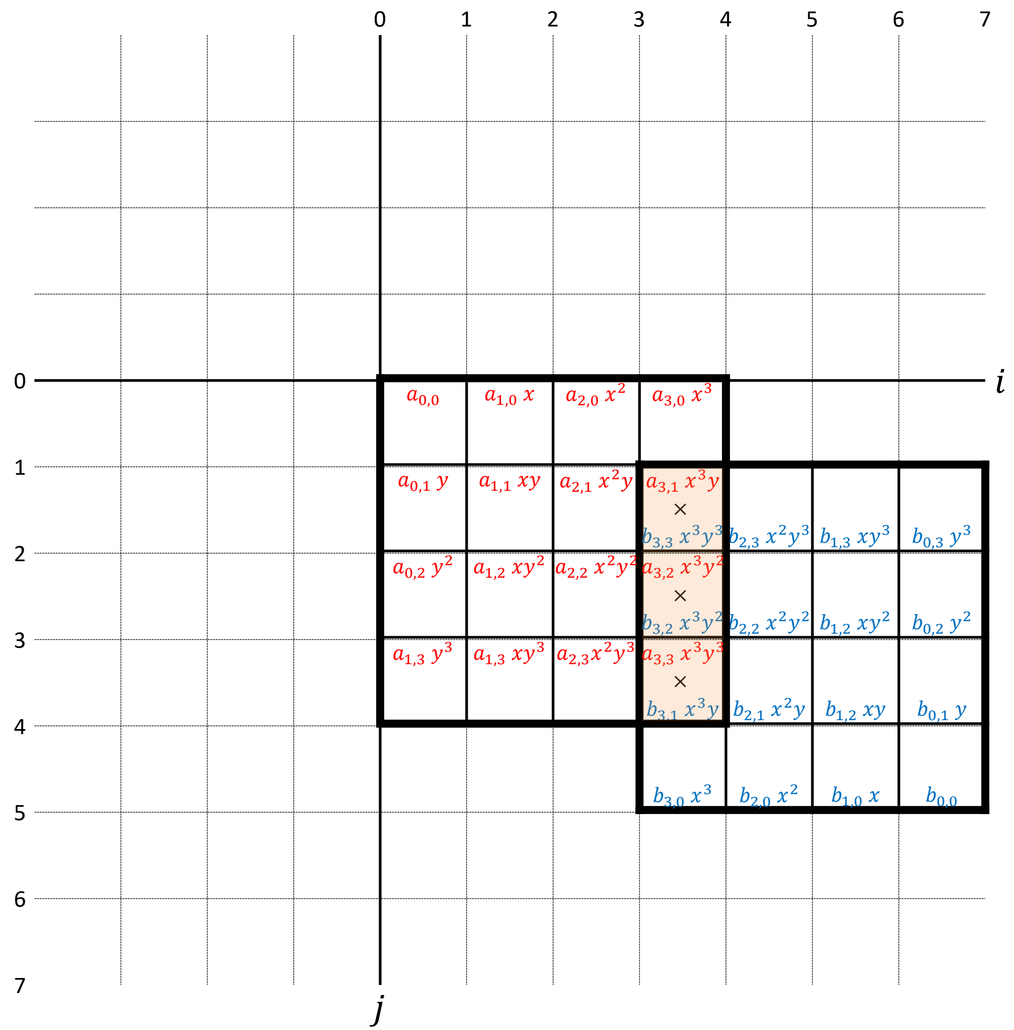
Coefficient of $x^{i-1}y^{j-1}$

Coefficient of x^0y^4

Multiplication of Bivariate Polynomials

Translation $\langle i, j \rangle$

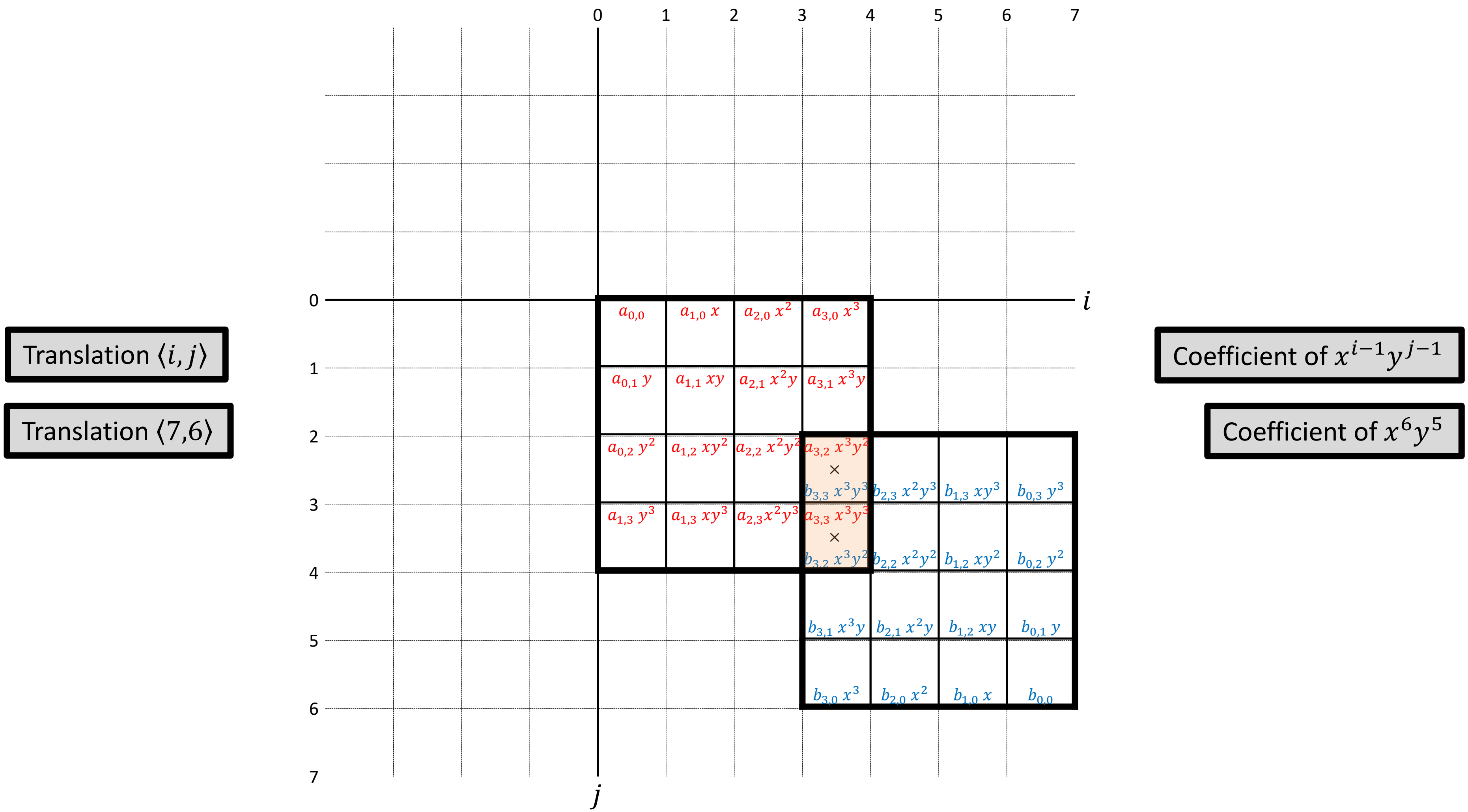
Translation $\langle 7{,}5 \rangle$



Coefficient of $x^{i-1}y^{j-1}$

Coefficient of x^6y^4

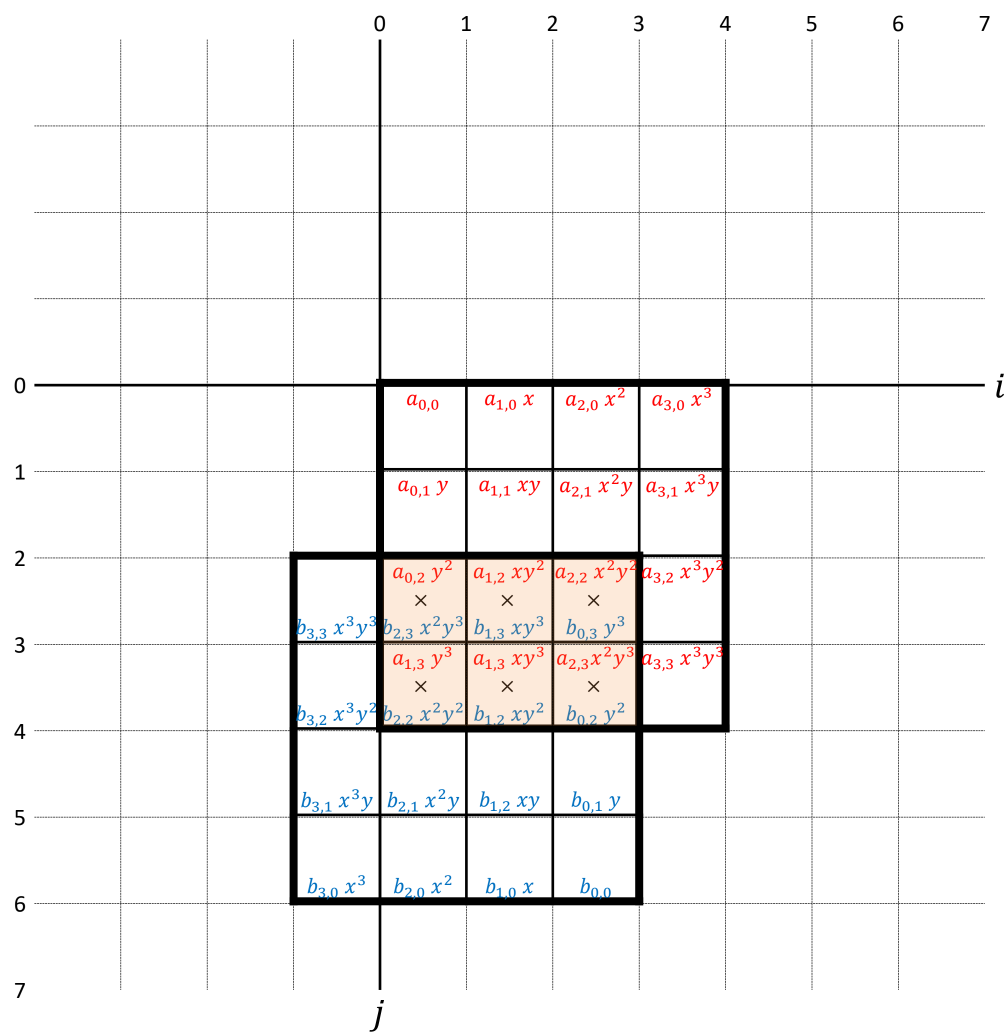
Multiplication of Bivariate Polynomials



Multiplication of Bivariate Polynomials

Translation $\langle i, j \rangle$

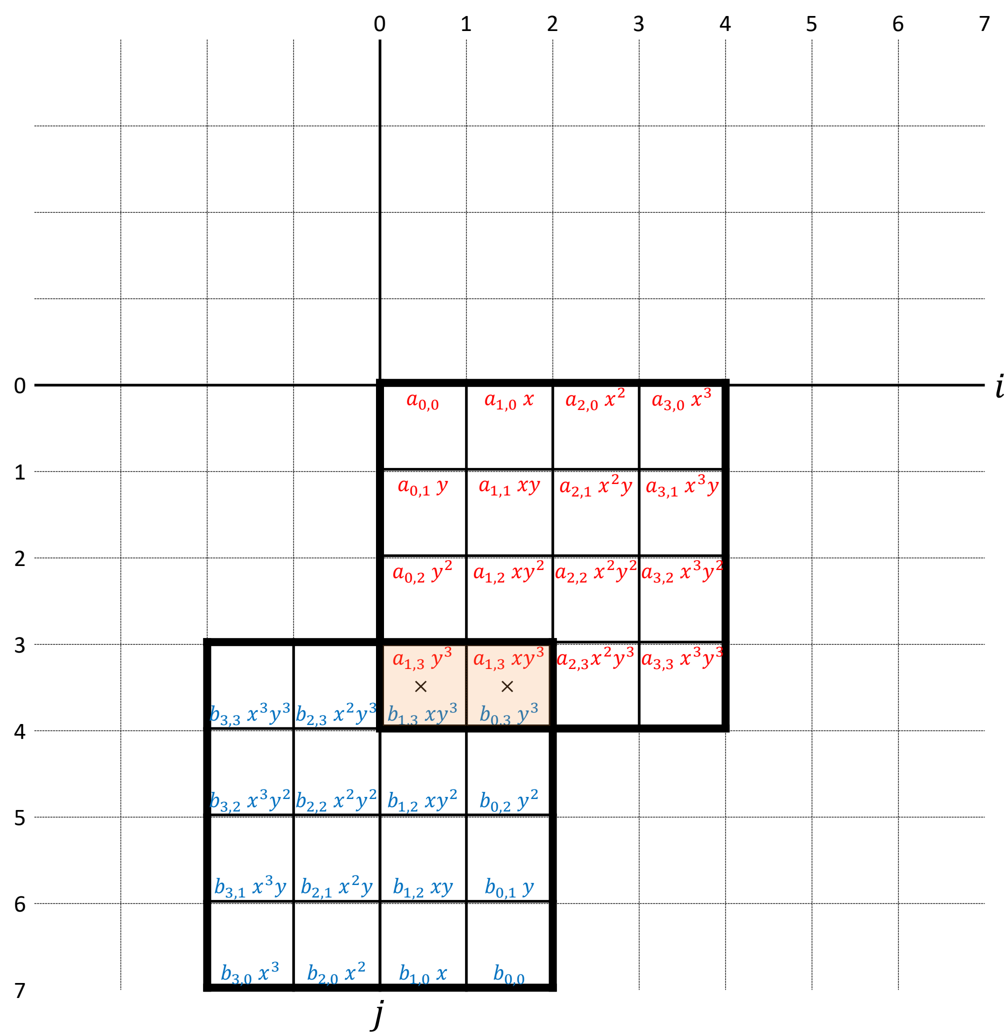
Translation $\langle 3,6 \rangle$



Coefficient of $x^{i-1}y^{j-1}$

Coefficient of x^2y^5

Multiplication of Bivariate Polynomials



Translation $\langle i,j\rangle$

Translation $\langle 2,7\rangle$

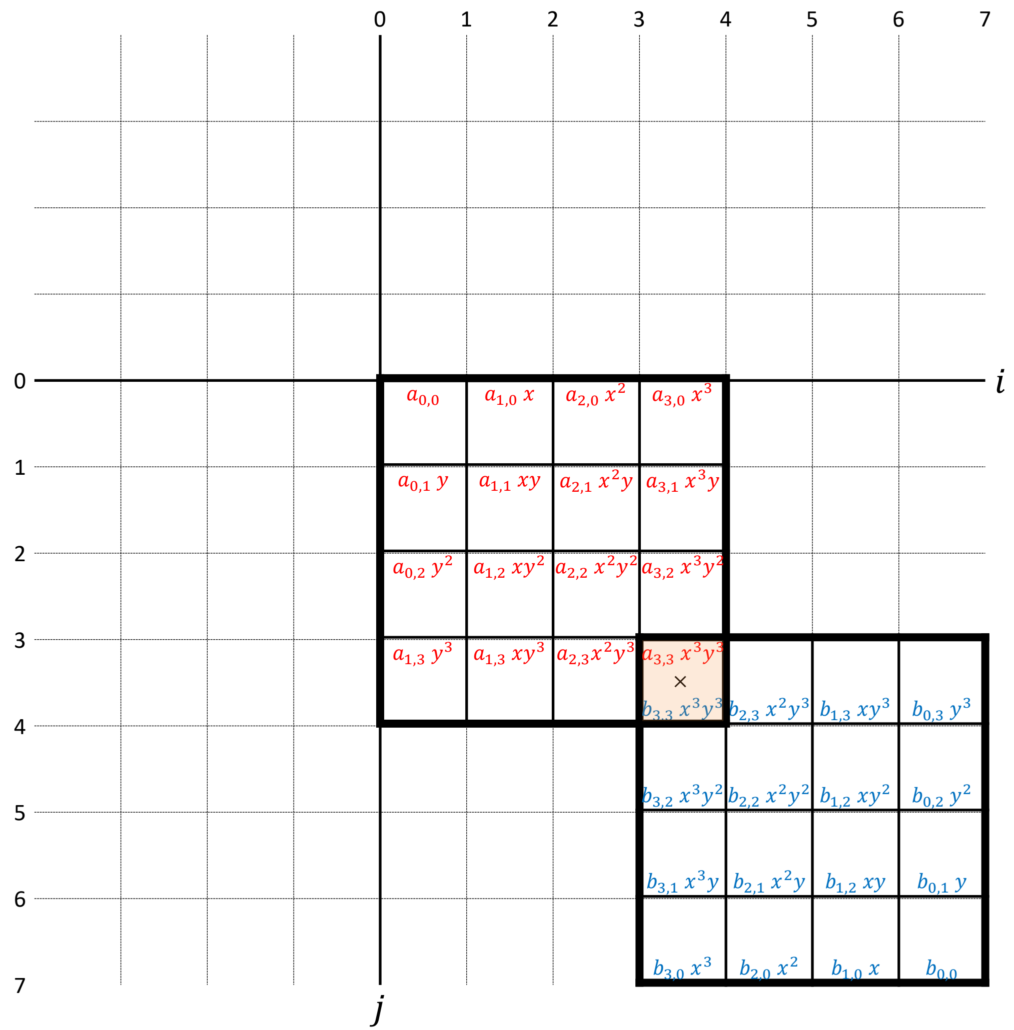
Coefficient of $x^{i-1}y^{j-1}$

Coefficient of x^1y^6

Multiplication of Bivariate Polynomials

Translation $\langle i,j\rangle$

Translation $\langle 7,7\rangle$



Coefficient of $x^{i-1}y^{j-1}$

Coefficient of x^6y^6