




分割統治法

計算機アルゴリズム特論：2015年度

只木進一



分割統治法(Divide and Conquer)と再帰

- 再帰で、よりサイズの小さい問題に分割
- 分割された、非常に小さい問題を解決する。
- その後、小さな問題の組み合わせを繰り返し、全体の問題を解く

分割統治法としてのMerge-Sort

- 長さ n のリストを、すでに整列済みの長さ $n/2$ のリストからの整列に、問題を分割
- 長さ $n/2$ の整列済みのリストからの整列の計算量：最悪で n
- 分割の回数 $\log_2 n$
 - 分割毎に n 個の要素の操作
- 計算量： $n \log_2 n$

最近接点：問題設定

- 二次元空間内の点の集合

$$P = \{p_k | 0 \leq k < n\}$$

- 最も距離の近い点の組を求める
- 単純に点の組の距離を調べたのでは、 $n \times (n - 1)/2$ 個の組について、調べる必要がある

最近接点：準備

- 全ての頂点を x 座標の順で並べたリスト X_0
- 全ての頂点を y 座標の順で並べたリスト Y_0
- それぞれ、 $n\log_2 n$ 回の比較で生成できる



点のリスト P



x 座標でソートしたリスト X_0

y 座標でソートしたリスト Y_0


Decorative wavy lines in the bottom left corner of the slide.

最近接点：分割

- 組 (X_i, Y_i) に対して、 X_i を左右 (X_{i+1}^L, X_{i+1}^R) に分割
- 対応して Y_i を割り当て、 (Y_{i+1}^L, Y_{i+1}^R) とする。

最近接点：分割

- 組 (X_i, Y_i) に対して、 Y_i を上下 (Y_{i+1}^D, Y_{i+1}^U) に分割
- 対応して X_i を割り当て、 (X_{i+1}^D, X_{i+1}^U) とする。
- 注：いずれのリストもソートされている



x 座標でソートしたリスト X_0



左右に分割

X_1^L

X_1^R

対応付け

Y_1^L

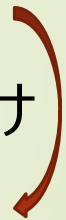
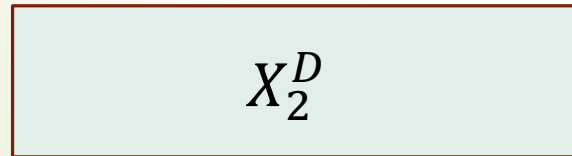
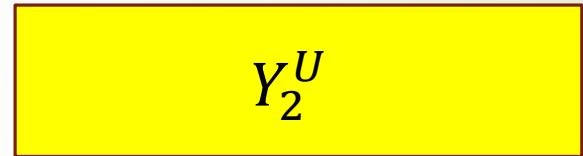
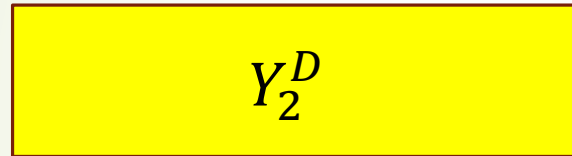
Y_1^R

y 座標でソートしたリスト Y_0

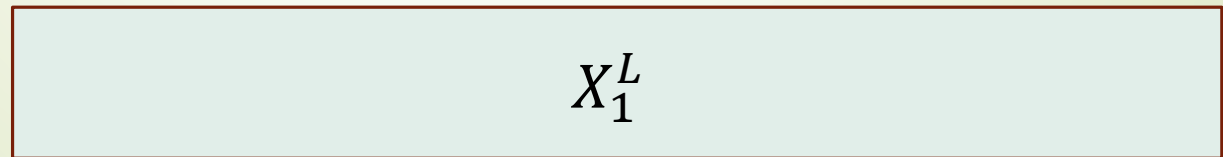
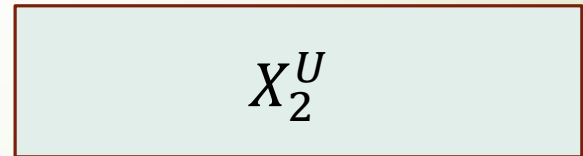




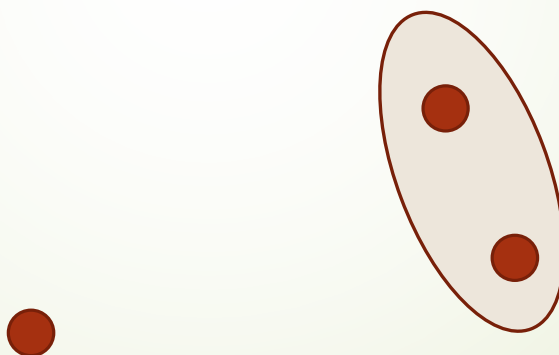
上下に分割

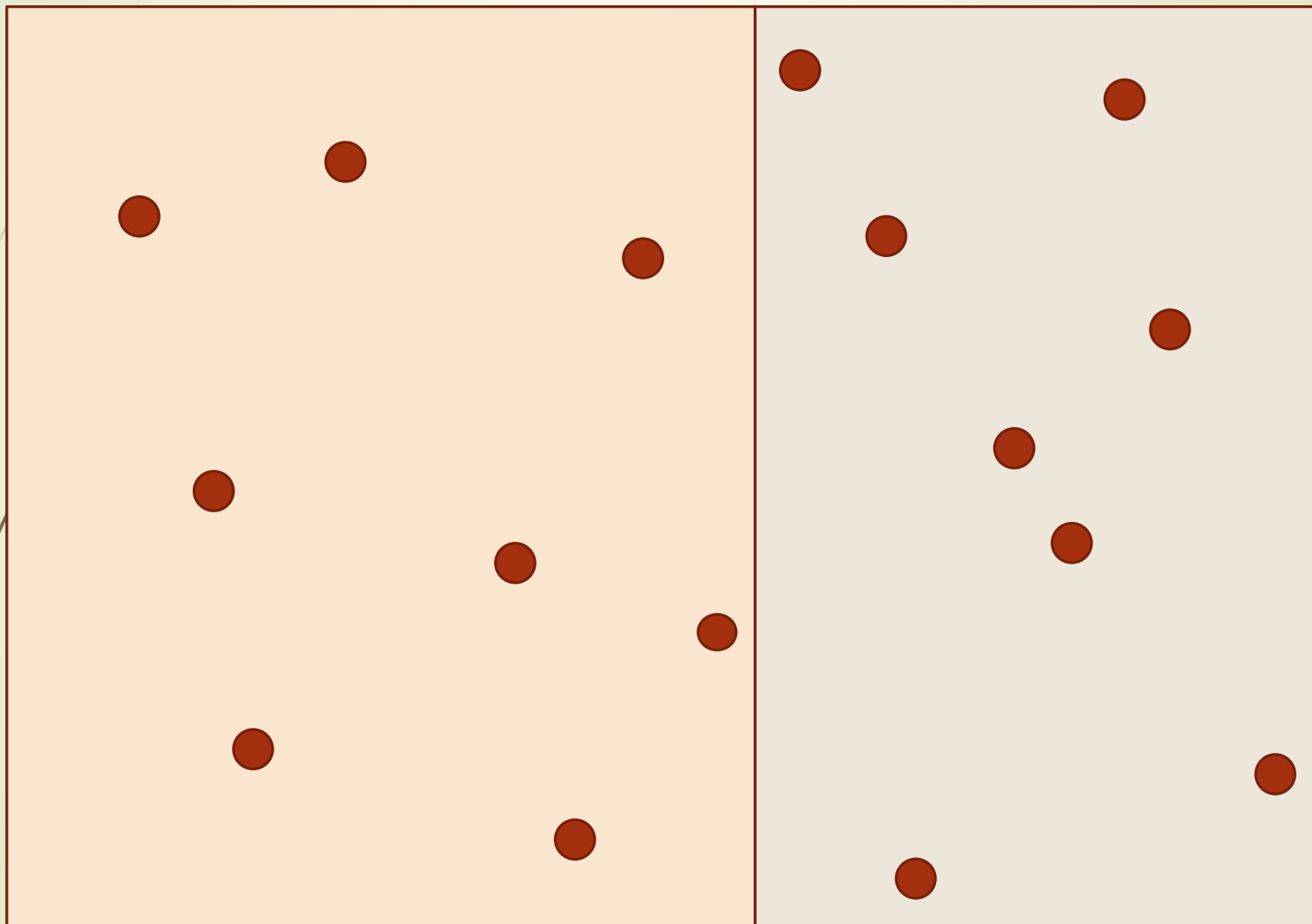


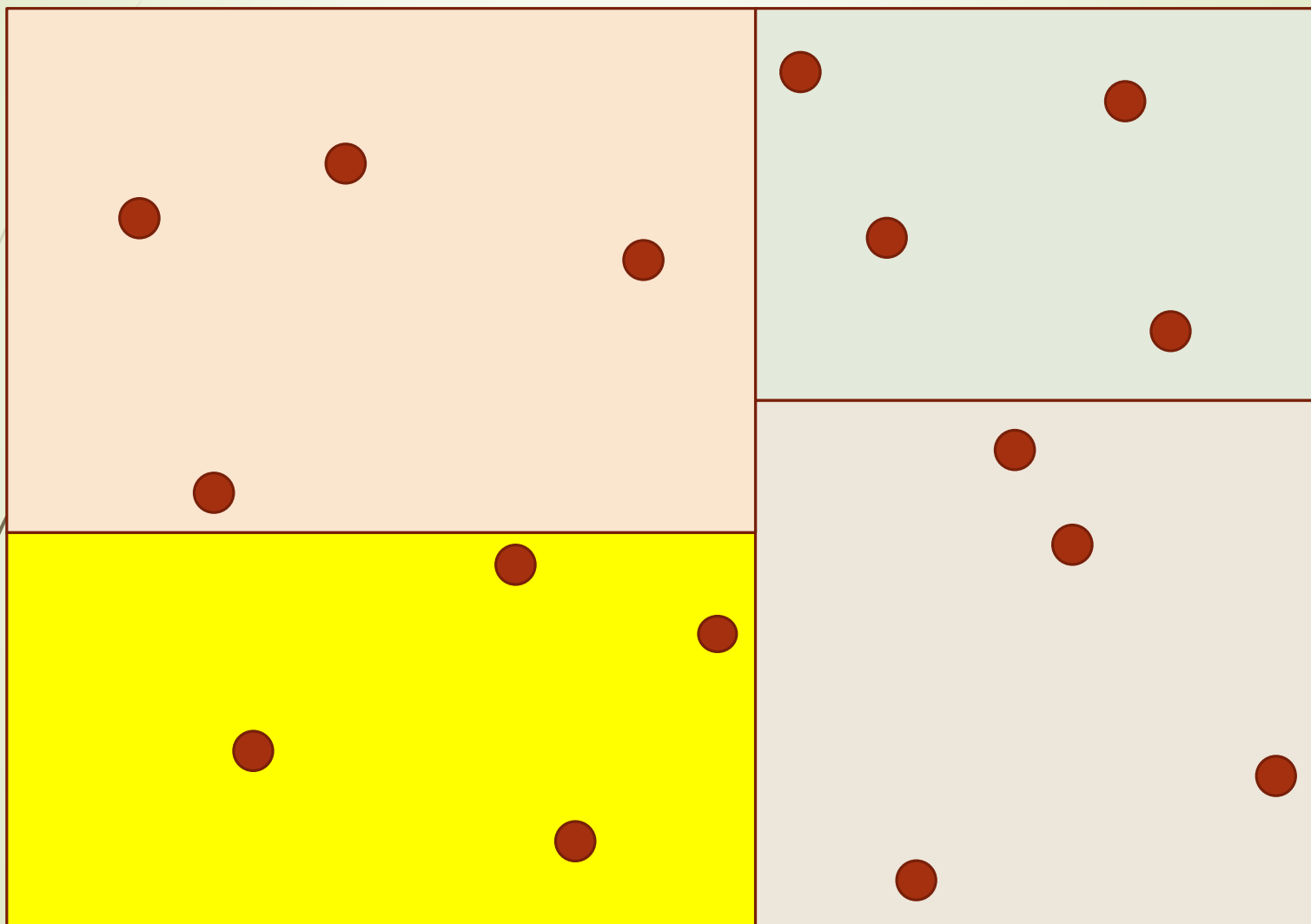
対応付け

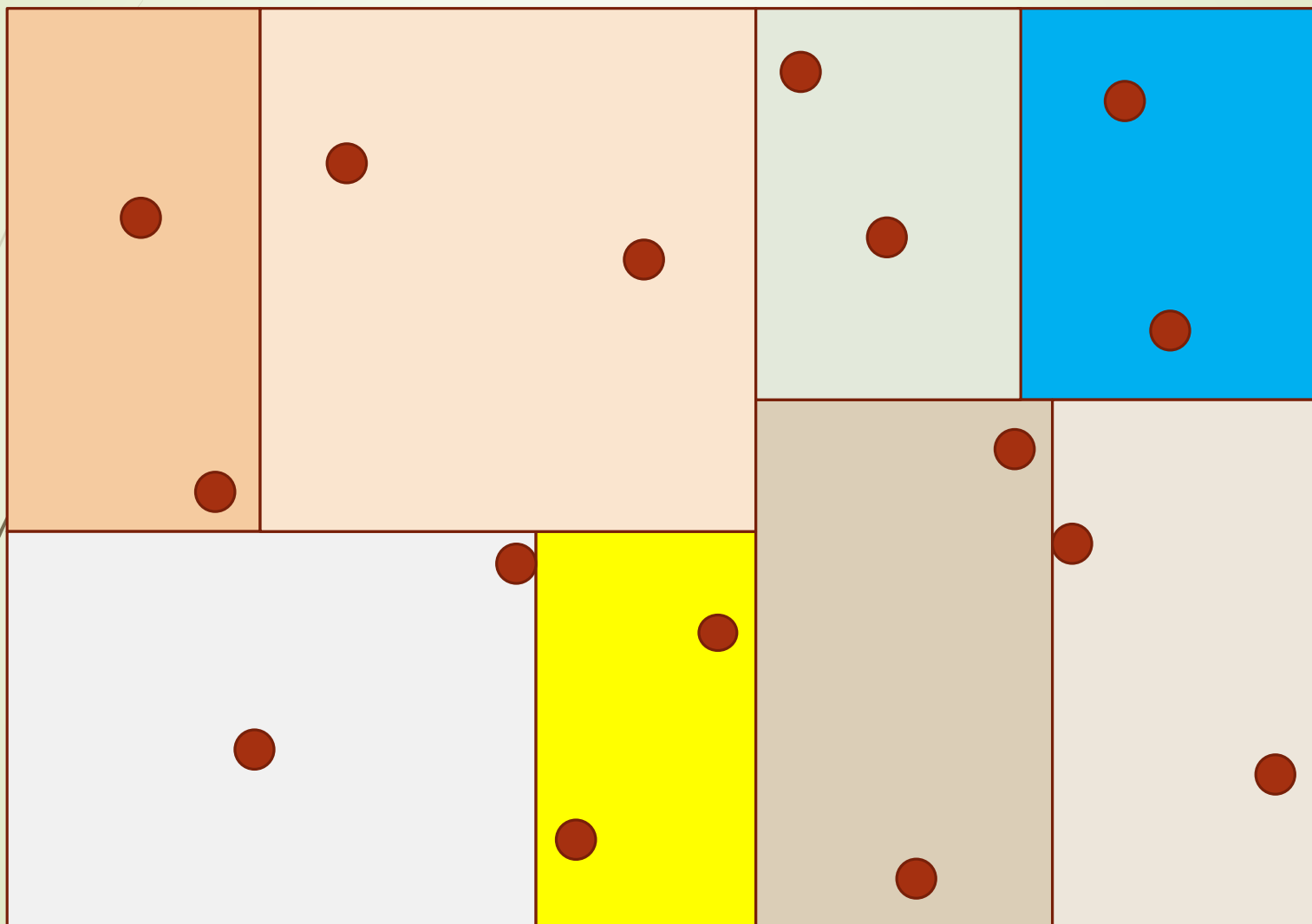


- 含まれている点の数が3以下になったら、最短距離の点の組を返す。







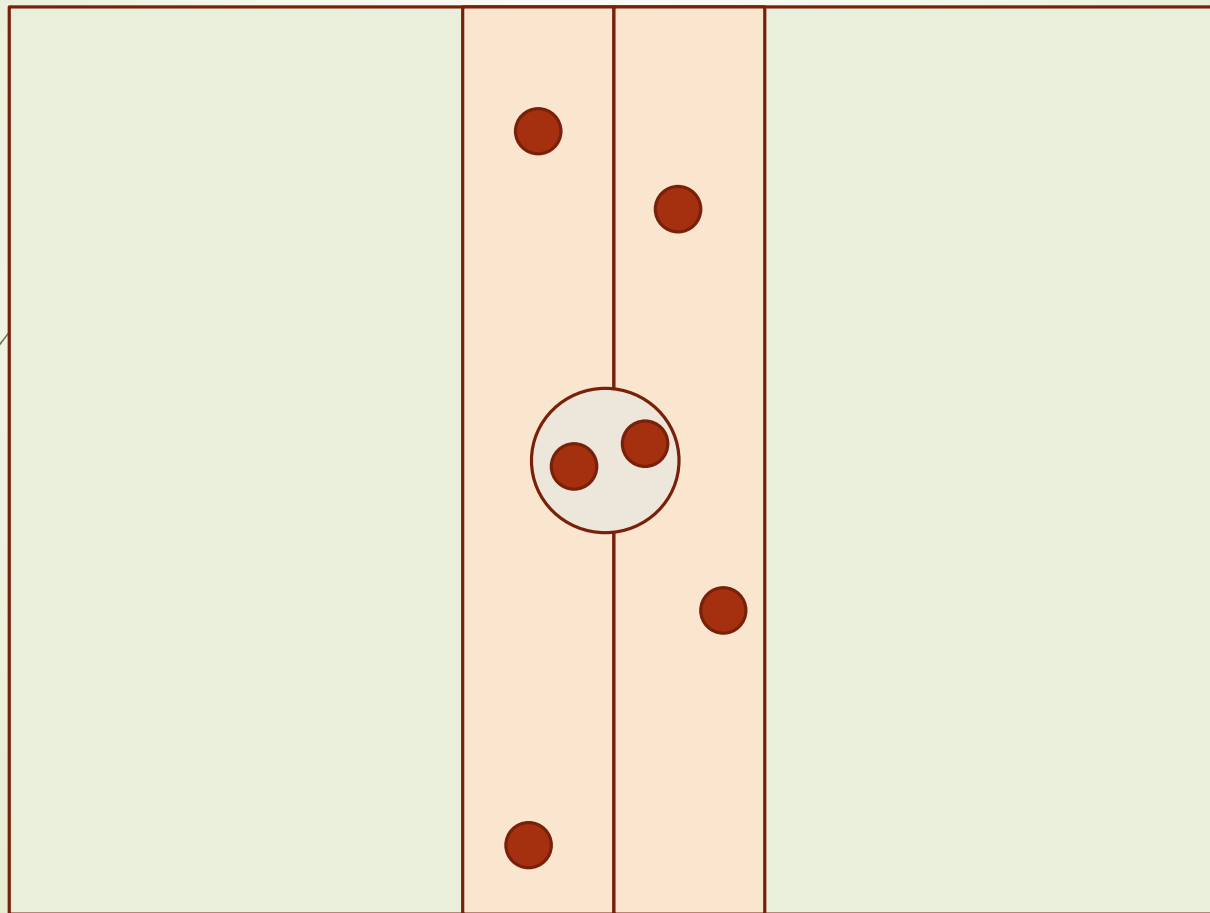


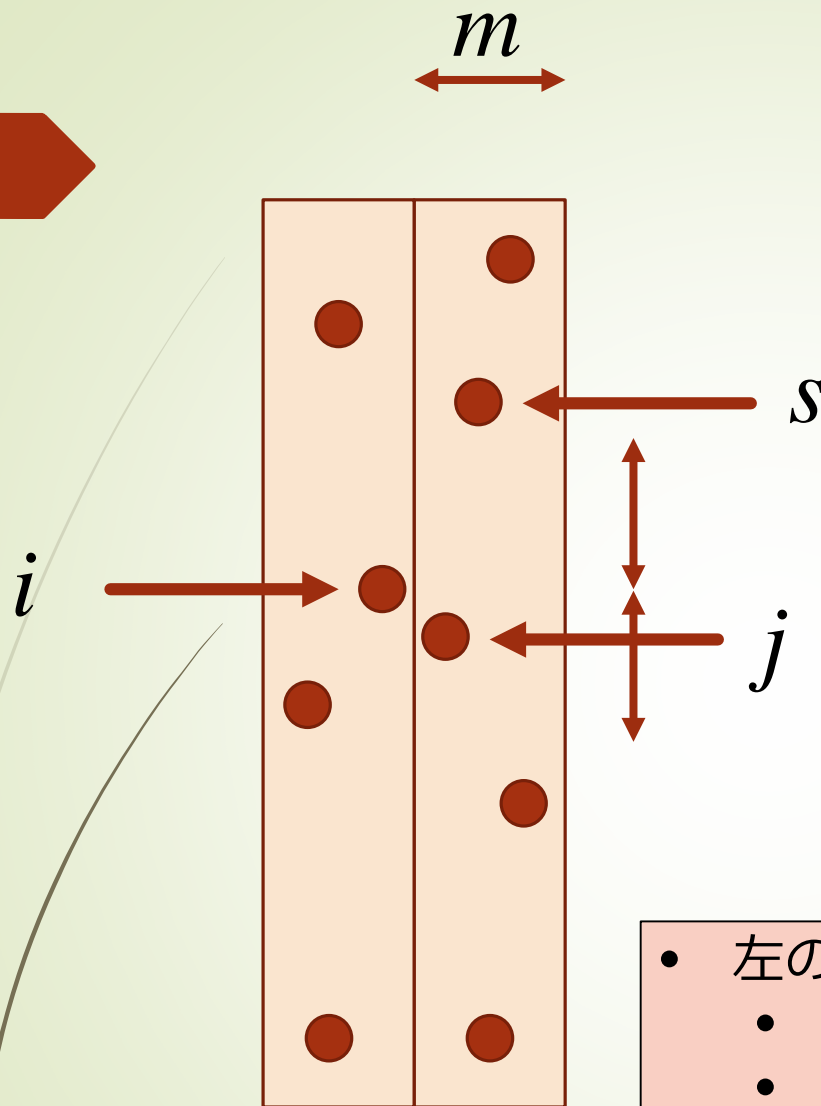
最近接点：統合

- 二つに分割した空間内のそれぞれの最近節点の組 p と q を得る。
- p が q より、距離が短いとし、その距離を m とする。
- 境界から距離 m 以内の点について、最近接点を求める

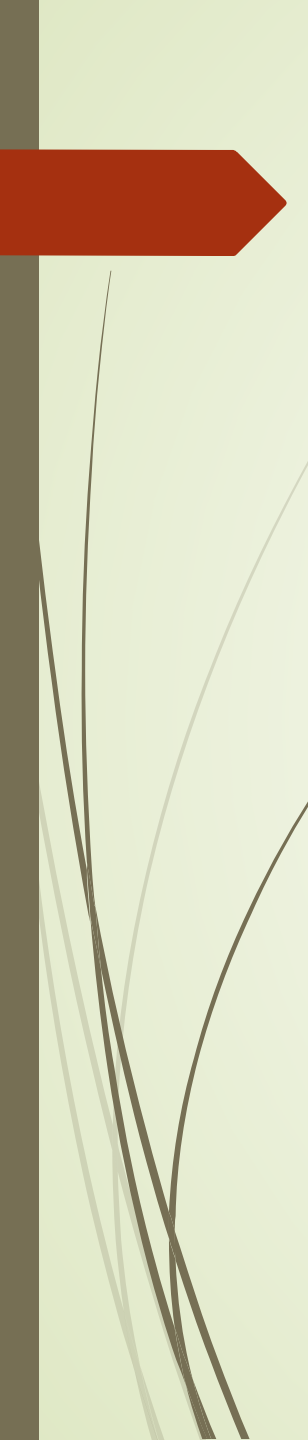


m

A horizontal double-headed arrow with a brown line and arrowheads at both ends, positioned below the letter m .



- 左のリストの各位置 i に対して
 - 右のリストとの距離を調査
 - s から開始
 - 下に y 座標が m 離れるまで
- s の更新
 - 始めて上に y 座標が m 以内に入った時



```

PointPair findMin(左リスト  $l$ , 右リスト  $r$ , 既知の組  $p$ ) {
     $d_{\min} = p$  の距離
     $p_{\text{new}} = p$ ; // 点の組
     $s = 0$ ; // 右リストの開始点
    forall(左リストの点  $p$ ) {
         $s_{\text{start}} = s$ ;  $j = s$ ;
        while ( $r_{j,y} < p_y + d_{\min}$ ) {
             $q = r_j$ ;
            if ( $q_y > p_y - d_{\min}$  &&  $s_{\text{start}} == s$ ) {  $s = j$ ; }
            if (distance( $q, p$ ) <  $d_{\min}$ ) { // 最短点の組を発見
                 $d_{\min} = \text{distance}(q, p)$ ;
                 $p_{\text{new}}$  に  $p$  と  $q$  を登録
            }
             $j++$ ;
            if ( $j \geq |r|$ ) { // 右リスト終了
                break;
            }
        }
    }
    return  $p_{\text{new}}$ ;
}

```