Algorithm closest_pair(p, n)
    MergeSort(p, 1, n) // Sort by x-axis
    return rec_cl_pair(p, 1, n)

Algorithm rec_cl_pair(p, i, j)
    if (j - i < 3)
        mergeSort(p, i, j) // Sort by y-axis
        delta ← dist(p[i], p[i + 1])
        if (j - i) = 1
            return delta
        if dist(p[i + 1], p[i + 2]) < delta
            delta ← dist(p[i + 1], p[i + 2])
        if dist(p[i], p[i + 2]) < delta
            delta ← dist(p[i], p[i + 2])
        return delta
    k ← (i + j)/2
    l ← p[k].x
    deltaL ← rec_cl_pair(p, i, k)
    deltaR ← rec_cl_pair(p, k + 1, j)
    delta ← min(deltaL, deltaR)
    merge(p, i, j, k) // merge sorted p[i]…p[k] with sorted p[k+1]…p[j]
    t ← 0
    for k ← i to j
        if (p[k].x > l - delta) && (p[k].x < l + delta)
            t ← t + 1
            v[t] ← p[k]
    for k ← 1 to t - 1
        for s ← k + 1 to min(t, k + 7)
            delta ← min(delta, dist(v[k], v[s]))
    return delta