12/08 週作業 p1

設計一個類別 MyPoint 可以記錄一個二維資料點 x, y 座標，並且提供計算與另一個二維資料點的距離函式 dist()。設計一個 MyPointSet 類別，其中 MyPoint 是它的一個內部類別，一個MyPointSet 物件可以記錄(包含)多個MyPoint 物件，，並且提供以下三種方式，計算二個MyPointSet 物件 A 與 B 彼此的距離：
(a)complete linkage (completeLink()): 找出所有可能的 MyPoint 物件組合 (a, b), 其中 a屬於A, b屬於 B, 回傳最大的距離 max(a.dist(b))
(b)single linkage(singleLink()): 如(a), 但回傳最小的距離 min(a.dist(b))
(c)average linkage: 如(a), 但回傳平均距離 sum(a.dist(b))/(n*m), 其中 n 為 A 包含的 MyPoint 物件數量，m為 B 包含的 MyPoint 物件數量

主程式的程式碼需能展示上述功能，例如：

```
:
MyPoint ptA[], ptB[];
MyPointSet A, B;
:
:
System.out.println("distance between ptA[0] and ptB[3] is "+ptA[0].dist(ptB[3]);
:
System.out.println("complete linkage of A and B is ", A.completeLink(B));
System.out.println("single linkage of B and A is "+ B.singleLink(A));
System.out.println("average linkage of A and B is "+A.averageLink(B));
:
```

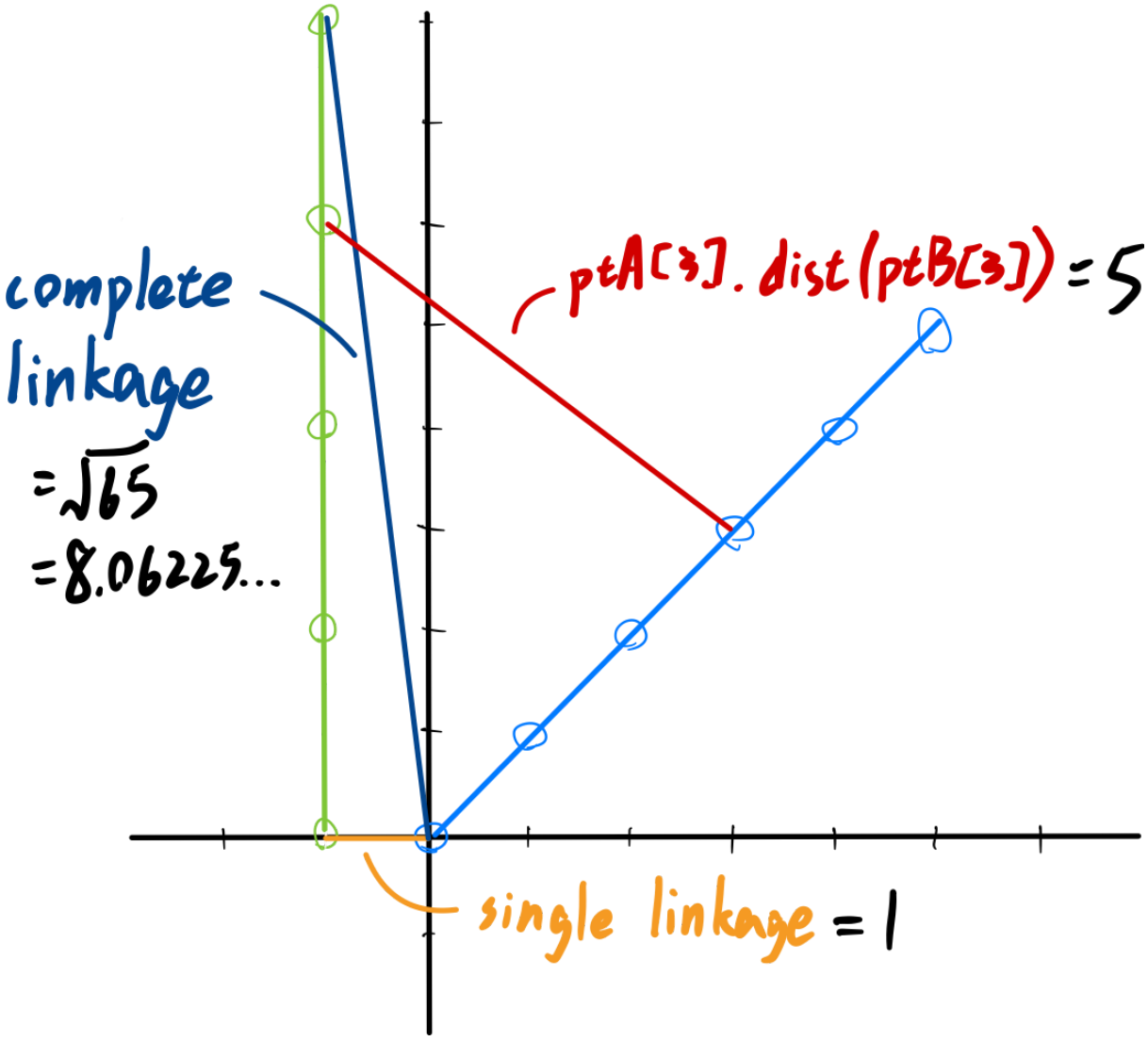
```
/* 12/08 楊育哲
 * 實作第一題: 點集合的距離計算(三種)
 */
public class h1_1208_w {
    static class MyPoint{
        private int x;
        private int y;
        MyPoint(int x_, int y_){this.x=x_; this.y=y_;}
        public double dist(MyPoint b){
            return Math.sqrt(Math.pow((x-b.x), 2)+Math.pow((y-b.y), 2));
        }
    }
    static class MyPointSet{
        private MyPoint[] set;
        private int points;
        MyPointSet(MyPoint[] a){
            points = a.length;
            set = a;
        }
        public double completeLink(MyPointSet B){
            double ans=0;
            for(int i=0; i<points; i++){
                for(int j=0; j<B.points; j++){
                    double temp = set[i].dist(B.set[j]);
                    if(ans<temp) ans=temp;
                }
            }
            return ans;
        }
        public double singleLink(MyPointSet B){
            double ans = 1000000;// or ans=completeLink(MyPoint B); 這樣比較保險，但時間會較久
            for(int i=0; i<points; i++){
                for(int j=0; j<B.points; j++){
                    double temp = set[i].dist(B.set[j]);
                    if(ans>temp) ans=temp;
                }
            }
            return ans;
        }
        public double averageLink(MyPointSet B){
            double ans=0;
            for(int i=0; i<points; i++){
                for(int j=0; j<B.points; j++){
                    ans+=set[i].dist(B.set[j]);
                }
            }
```

```
            return ans/(points*B.points);
        }
        void view(){
            for(int i=0; i<points; i++)
                System.out.printf("(%d, %d), ", set[i].x, set[i].y);
            System.out.printf("points: %d\n", points);
        }
    }
    static public void main(String args[]){
        MyPoint[] ptA = new MyPoint[5];
        MyPoint[] ptB = new MyPoint[5];
        for(int i=0; i<5; i++) ptA[i]=new MyPoint(i, i);
        for(int i=0; i<5; i++) ptB[i]=new MyPoint(-1, i*2);
        MyPointSet A = new MyPointSet(ptA);
        MyPointSet B = new MyPointSet(ptB);
        System.out.println("A and ptA[] information:");
        A.view();
        System.out.println("B and ptB[] information:");
        B.view();

        System.out.printf("ptA[3].dist(ptB[3]) = %f\n", ptA[3].dist(ptB[3]));//(3, 3) <-> (-1, 6) dist=5

        System.out.println("complete linkage of A and B is "+ A.completeLink(B)); //根號(64+1): (0, 0) , (-1, 8)
        System.out.println("single linkage of B and A is "+ B.singleLink(A));// 1: (-1, 0) , (0, 0)
        System.out.println("average linkage of A and B is "+ A.averageLink(B));
    }
}
```

輸出：

A and ptA[] information:
(0, 0), (1, 1), (2, 2), (3, 3), (4, 4), points: 5
B and ptB[] information:
(-1, 0), (-1, 2), (-1, 4), (-1, 6), (-1, 8), points: 5
ptA[3].dist(ptB[3]) = 5.000000
complete linkage of A and B is 8.06225774829855
single linkage of B and A is 1.0
average linkage of A and B is 4.69569503479292

說明：


complete
linkage
=√65
=8.06225...
ptA[3].dist(ptB[3])=5
single linkage=1