演算法 Homework 20190926
山下夏輝（Yamashita Natsuki）
R08922160

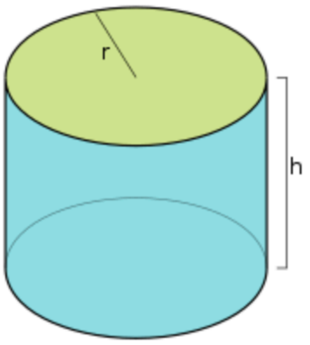
Mini HW #2

截止時間 星期四，14:20 **總分** 1 **繳交** 線上輸入或者檔案上傳
檔案類型 pdf **接受繳交時間** 9月26日 14:20 - 10月3日 14:20 7天

Given N points on a cylindrical surface, please design an $O(N \lg N)$ algorithm to find the closest pair of the N points. Note that you should:
(1) (40%) Write pseudo-code to describe the steps
(2) (30%) Justify the correctness of your algorithm
(3) (30%) Prove its time complexity by using recursion-tree method.

* The distance between two points on a cylindrical surface is defined as the length of the shortest path **along the surface**.

* A cylindrical surface is the blue-shaded surface in the attach picture.



//preparing

    //n = (x, y, z) // origin is the center of the cylinder's bottom.
    //   = (r cos θ, r sinθ, h)
    //|n_1n_2| = $\sqrt{(x1 - x2)^2 + (y1 - y2)^2 + (z1 - z2)^2}$

//reference

    https://www.youtube.com/watch?v=_nPvJpjKBmw
    slides of the classes
    guidance of TA.吳士綸

(1)

```
Pre-Processing(P, N)
    //sort
    Sort P by x-, y-, z-coordinate and store in Px, Py and Pz
Closest-Pair(P, N)
    //base case
    If |N| <= 3
        return Brute-Force-ClosestPair-and-Distance(P)
    //recursive case
    //divide
    Find a vertical line L s.t.both planes contain half of the points
    //conquer
    left-min = Closest-Pair(points in the left, the num of the points in the left)
    right-min = Closest-Pair(Points in the right, the num of the points in the right)
    //combine
    delta = min(left-min, right-min)
    //cross two regions case
    Remove points that are delta or more away from L
    for point $p_i$ in candidates sorted by y-coordinate.
        compute distances with $p_{i+1}$, $p_{i+2}$, ⋯, $p_{i+15}$
        update delta if a closer pair is found
    return the closest pair and its distance
```

(2)

Proof

In base case, $|N| \leq 3$, the answer is returned by the brute force algorithm

In recursive case, $|N| > 4$, Closest-Pair(P,N) will be recalled recursively. The answer is in the variable delta and will be returned at last when the closest pair is not crossing two region divided by L.

In recursive case and in cross two regions case, $|N| > 4$, the value of delta will be updated when a closer pair is found and the answer is in the variable delta and will be returned at last.

(3)

Time Complexity for Finding Closest Pair

```
    Pre-Processing(P, N)
        sort:                   O(nlogn)
    Closest-Pair(P, N)
        base case:              O(1)
        divide:                 O(n)
        conquer:                2T(n/2)
```

combine:	$O(1)$
cross two regions case:	$O(n)$

$T(n) = 2T(n/2)+O(n\log n)+O(n)+O(n)$

$= 2T(n/2) + O(n\log n)$

Theorem

$$T(n) = \begin{cases} O(1) & \text{if } n\leq 3 \\ 2T(n/2) + O(n) & \text{if } n>3 \end{cases}$$

$T(n) = 2T(n/2) + cn$

≤ 1

$\leq 2[2T(n/4)+cn/2]+cn \quad = 4T(n/4)+2cn$

$\leq 4[2T(n/8)+cn/4]+2cn \quad = 8T(n/8)+3cn$

…

$\leq 2^kT(n/2^k)+kcn$

The expansion stops when $2^k = n$.

Then,

$T(n) \leq nT(1)+cn\log n$

$=O(n) + O(n\log n)$

$= O(n\log n)$

Also, the complexity of the pre-processing sort algorithm is $O(n\log n)$.

So, it can be proved that the complexity of this algorithm is $O(n\log n)$.