

**清华**

1. 求面积:  
 $\text{Left}(p, q, s):$   
 $\text{Area}(p, q, s) > 0$   
 (Area=0, 说明pq共线)  
 $\text{Area}(p, q, s):$   
 $2 \cdot \text{SpqS} = \begin{vmatrix} p_x & p_y & 1 \\ q_x & q_y & 1 \\ s_x & s_y & 1 \end{vmatrix}$   
 2. 面积内: ~~Area > 0~~  $\text{Left}$   
 3. 面积为0:  $\text{Left}$   
 4. 面积为负:  $\text{Left}$

① 线性求二者的 kernel  $\Rightarrow$  并序向交

若,

(

- ① 按相临顺序找段
- ② 线性双向遍历 (~~被~~ 左段 + 右段)
- ③ 每次加入新线段, 把 ~~被~~ (左端 所有右端) 更新
- ④ ~~更新左端~~

最后用左端线段处理阶段, 得到封闭 kernel  $O(n \log n)$

性质:

①  $\cong \mathbb{Z}$ ,

A graph with four vertices labeled p, q, r, and s. Vertex p is at the top, q is at the bottom, r is on the left, and s is on the right. Edges connect p to q, p to r, p to s, q to r, and q to s. There are also edges extending from r to the left and from s to the right.

$y = ax - b$


A handwritten graph showing a network of nodes and edges. The nodes are labeled  $P(9)$ ,  $P(5)$ ,  $P(10)$ ,  $P(15)$ , and  $P(1)$ . There are also labels  $P(9)$  and  $P(10)$  above the nodes. A curved arrow points from the left towards the node  $P(5)$ .

证明: ① MAT ~~是~~  
图灵可识别,  
给定任意的  $p, q$ , 构造图灵机  $M$  如下:  
是  $p, q$  是否相等  
↓  
构造  $D(p)$  与  $D(q)$  交集  $S$   
 $O(n^2)$

① 問答

↑  
○  
○  
↓

(f k p y m) 5  
~~問答~~

② Lower Hull & Upper envelope  
Lower Hull: 

PELHCP'E 2E700h 2E700h

(supr. enabr : 在所有底上的反A

0123: Over Hall

$01 \rightarrow 12 \rightarrow 23$ : upper envelope

























① Sweep line 扫线































































① monobre poly. 对字信号, 延迟控制(有序)

年成: 自西向东向西

1. 海空联运  
 2. 陆空联运  
 3. 陆海联运  
 4. 陆陆联运  
 5. 空空联运  
 6. 海海联运  
 7. 海陆联运  
 8. 陆海联运  
 9. 陆陆联运  
 10. 空空联运  
 11. 海海联运  
 12. 海陆联运  
 13. 陆海联运  
 14. 陆陆联运  
 15. 空空联运  
 16. 海海联运  
 17. 海陆联运  
 18. 陆海联运  
 19. 陆陆联运  
 20. 空空联运  
 21. 海海联运  
 22. 海陆联运  
 23. 陆海联运  
 24. 陆陆联运  
 25. 空空联运  
 26. 海海联运  
 27. 海陆联运  
 28. 陆海联运  
 29. 陆陆联运  
 30. 空空联运  
 31. 海海联运  
 32. 海陆联运  
 33. 陆海联运  
 34. 陆陆联运  
 35. 空空联运  
 36. 海海联运  
 37. 海陆联运  
 38. 陆海联运  
 39. 陆陆联运  
 40. 空空联运  
 41. 海海联运  
 42. 海陆联运  
 43. 陆海联运  
 44. 陆陆联运  
 45. 空空联运  
 46. 海海联运  
 47. 海陆联运  
 48. 陆海联运  
 49. 陆陆联运  
 50. 空空联运  
 51. 海海联运  
 52. 海陆联运  
 53. 陆海联运  
 54. 陆陆联运  
 55. 空空联运  
 56. 海海联运  
 57. 海陆联运  
 58. 陆海联运  
 59. 陆陆联运  
 60. 空空联运  
 61. 海海联运  
 62. 海陆联运  
 63. 陆海联运  
 64. 陆陆联运  
 65. 空空联运  
 66. 海海联运  
 67. 海陆联运  
 68. 陆海联运  
 69. 陆陆联运  
 70. 空空联运  
 71. 海海联运  
 72. 海陆联运  
 73. 陆海联运  
 74. 陆陆联运  
 75. 空空联运  
 76. 海海联运  
 77. 海陆联运  
 78. 陆海联运  
 79. 陆陆联运  
 80. 空空联运  
 81. 海海联运  
 82. 海陆联运  
 83. 陆海联运  
 84. 陆陆联运  
 85. 空空联运  
 86. 海海联运  
 87. 海陆联运  
 88. 陆海联运  
 89. 陆陆联运  
 90. 空空联运  
 91. 海海联运  
 92. 海陆联运  
 93. 陆海联运  
 94. 陆陆联运  
 95. 空空联运  
 96. 海海联运  
 97. 海陆联运  
 98. 陆海联运  
 99. 陆陆联运  
 100. 空空联运

[illegible]

helper will: ~~right helper (j) is L~~  
 0: info (1)   
  
 1: info (1)   
 2: info (1)   
 3: info (1)   
 4: info (1)   
 5: info (1)   
 6: info (1)   
 7: info (1)   
 8: info (1)   
 9: info (1)   
 10: info (1)   
 11: info (1)   
 12: info (1)   
 13: info (1)   
 14: info (1)   
 15: info (1)   
 16: info (1)   
 17: info (1)   
 18: info (1)   
 19: info (1)   
 20: info (1)   
 21: info (1)   
 22: info (1)   
 23: info (1)   
 24: info (1)   
 25: info (1)   
 26: info (1)   
 27: info (1)   
 28: info (1)   
 29: info (1)   
 30: info (1)   
 31: info (1)   
 32: info (1)   
 33: info (1)   
 34: info (1)   
 35: info (1)   
 36: info (1)   
 37: info (1)   
 38: info (1)   
 39: info (1)   
 40: info (1)   
 41: info (1)   
 42: info (1)   
 43: info (1)   
 44: info (1)   
 45: info (1)   
 46: info (1)   
 47: info (1)   
 48: info (1)   
 49: info (1)   
 50: info (1)   
 51: info (1)   
 52: info (1)   
 53: info (1)   
 54: info (1)   
 55: info (1)   
 56: info (1)   
 57: info (1)   
 58: info (1)   
 59: info (1)   
 60: info (1)   
 61: info (1)   
 62: info (1)   
 63: info (1)   
 64: info (1)   
 65: info (1)   
 66: info (1)   
 67: info (1)   
 68: info (1)   
 69: info (1)   
 70: info (1)   
 71: info (1)   
 72: info (1)   
 73: info (1)   
 74: info (1)   
 75: info (1)   
 76: info (1)   
 77: info (1)   
 78: info (1)   
 79: info (1)   
 80: info (1)   
 81: info (1)

$e_L$  /  /  $e_R$   
 = helper; (1)  /  $e_L$  /  $e_R$   
 (2)  /  $e_L$  /  $e_R$   
 (3) stalactite /  /  $e_L$  /  $e_R$   
 (4)  /  $e_L$  /  $e_R$   
 (5)  /  $e_L$  /  $e_R$   
 (6)  /  $e_L$  /  $e_R$   
 (7)  /  $e_L$  /  $e_R$   
 (8)  /  $e_L$  /  $e_R$   
 (9)  /  $e_L$  /  $e_R$   
 (10)  /  $e_L$  /  $e_R$   
 (11)  /  $e_L$  /  $e_R$   
 (12)  /  $e_L$  /  $e_R$   
 (13)  /  $e_L$  /  $e_R$   
 (14)  /  $e_L$  /  $e_R$   
 (15)  /  $e_L$  /  $e_R$   
 (16)  /  $e_L$  /  $e_R$   
 (17)  /  $e_L$  /  $e_R$   
 (18)  /  $e_L$  /  $e_R$   
 (19)  /  $e_L$  /  $e_R$   
 (20)  /  $e_L$  /  $e_R$   
 (21)  /  $e_L$  /  $e_R$   
 (22)  /  $e_L$  /  $e_R$   
 (23)  /  $e_L$  /  $e_R$   
 (24)  /  $e_L$  /  $e_R$   
 (25)  /  $e_L$  /  $e_R$   
 (26)  /  $e_L$  /  $e_R$   
 (27)  /  $e_L$  /  $e_R$   
 (28)  /  $e_L$  /  $e_R$   
 (29)  /  $e_L$  /  $e_R$   
 (30)  /  $e_L$  /  $e_R$   
 (31)  /  $e_L$  /  $e_R$   
 (32)  /  $e_L$  /  $e_R$   
 (33)  /  $e_L$  /  $e_R$   
 (34)  /  $e_L$  /  $e_R$   
 (35)  /  $e_L$  /  $e_R$   
 (36)  /  $e_L$  /  $e_R$   
 (37)  /  $e_L$  /  $e_R$   
 (38)  /  $e_L$  /  $e_R$   
 (39)  /  $e_L$  /  $e_R$   
 (40)  /  $e_L$  /  $e_R$   
 (41)  /  $e_L$  /  $e_R$   
 (42)  /  $e_L$  /  $e_R$   
 (43)  /  $e_L$  /  $e_R$   
 (44)  /  $e_L$  /  $e_R$   
 (45)  /  $e_L$  /  $e_R$   
 (46)  /  $e_L$  /  $e_R$   
 (47)  /  $e_L$  /  $e_R$   
 (48)  /  $e_L$  /  $e_R$   
 (49)  /  $e_L$  /  $e_R$   
 (50)  /  $e_L$  /  $e_R$   
 (51)  /  $e_L$  /  $e_R$   
 (52)  /  $e_L$  /  $e_R$   
 (53)  /  $e_L$  /  $e_R$   
 (54)  /  $e_L$  /  $e_R$   
 (55)  /  $e_L$  /  $e_R$   
 (56)  /  $e_L$  /  $e_R$   
 (57)  /  $e_L$  /  $e_R$   
 (58)  /  $e_L$  /  $e_R$   
 (59)  /  $e_L$  /  $e_R$   
 (60)  /  $e_L$  /  $e_R$   
 (61)  /  $e_L$  /  $e_R$   
 (62)  /  $e_L$  /  $e_R$   
 (63)  /  $e_L$  /  $e_R$   
 (64)  /  $e_L$  /  $e_R$   
 (65)  /  $e_L$  /  $e_R$   
 (66)  /  $e_L$  /  $e_R$   
 (67)  /  $e_L$  /  $e_R$   
 (68)

2. Pelanray 三部分 (对数)

主因: ① Ecacilid 多生毒 (用这个药因怕(不洁))  
② Ecacilid TSP

性质: (空图): 任何力的外圈必为空白 (不含边) (反例)

递归:  $R(n)$  (注:  $T(a, b, c) = \text{TriangleContaining}(p, i, p)$ )  
 $O(n \log n)$

$\text{STest}(p, a, b)$   
 $x = \text{right\_sight}(a, b)$  if  $(!x)$  return  
 if  $(\text{in\_circle}(p, a, b, x))$   
 then  $p \leftrightarrow x$   
 $\text{STest}(p, a, x), \text{STest}(p, x, b)$

3. Volcani (多)  
(多量の火山)

姓在(古)的(上)好次(天)  
姓有(他)

夏桂屏: O(Nbgn) (ID: 113 (64 材料))  
2007

构型: ~~VII~~ ~~III~~ ~~II~~  
反式结构

① Divide: 按车厢平均, 在站前站地

② Meye: 自上而下地找出的 contour (特征)

① 求  $E$ , 荷有  $\int \rho$  的上下板  
② 求  $E_{S1}$ ,  $E_{S2}$ ,  $E_{SP}$  的场,  $b_1 \neq b_2$  (1) 板平齐.  
其  $b_1 \neq b_2$  (2) 板不平齐,  $b_1 \neq b_2$  (3) 板不平齐

① cell 1 在 (0,0) 的左上方  
 cell 2  
 ②  
 ③  
 ④  
 ⑤  
 ⑥  
 ⑦  
 ⑧  
 ⑨  
 ⑩  
 ⑪  
 ⑫  
 ⑬  
 ⑭  
 ⑮  
 ⑯  
 ⑰  
 ⑱  
 ⑲  
 ⑳  
 ㉑  
 ㉒  
 ㉓  
 ㉔  
 ㉕  
 ㉖  
 ㉗  
 ㉘  
 ㉙  
 ㉚  
 ㉛  
 ㉜  
 ㉝  
 ㉞  
 ㉟  
 ㊱  
 ㊲  
 ㊳  
 ㊴  
 ㊵  
 ㊶  
 ㊷  
 ㊸  
 ㊹  
 ㊺  
 ㊻  
 ㊼  
 ㊽  
 ㊾  
 ㊿

IP: incircle Test  $(a, b, c, p)$

$$\begin{vmatrix} a_x & d_y & ax^2+ay^2 \\ b_x & b_y & bx^2+by^2 \\ c_x & c_y & cx^2+cy^2 \\ p_x & p_y & px^2+py^2 \end{vmatrix} \quad 70 \text{ @ } p_{\text{total}} \text{ @ } 41.74 \text{ @ } \#$$

② DCL 数据块  
对于多次遍历  
(指针结构)

并也: 记录块 / turn  
inc / 1  
块 / 60  
指针结构 (inc)  
块: 记录块 / 块  
并也 inc  
块: 记录块

inc = 6  
inc a / turn  
pre  
inc  
a

[illegible]

7) Q1-L Delaunay's

原圖的長寬 = 高, 分

对价 (b) 1000 (d)



扫描全能王 创建



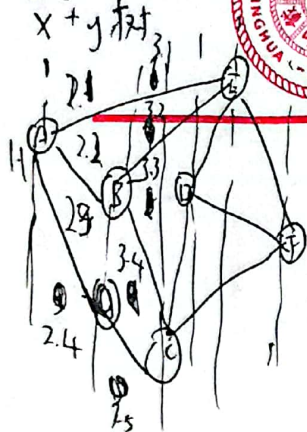
# 清华大学

线: 不相交, 非空直: Temporal Map

RL (树机面付输入) 并结

1. point location: 点定位

和  $x+y$  树

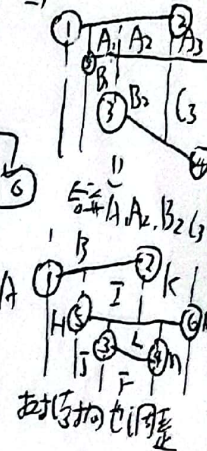
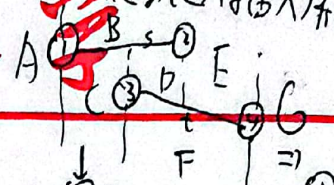


时间  $O(n \log n)$

空间  $O(n^2) \Rightarrow$  用 PPS

查找: 时间  $O(\log n)$  (Partially Persistent Structure)  
(仅读,  $O(n \log n)$ )

构建: 时间  $O(n \log n)$   
空间  $O(n)$   
查找:  $O(\log n)$



2. Range Search: 区间范围内的点

1D: 二叉搜索树 BST  $\Rightarrow$  中位数分裂 (时间  $O(n \log n)$ )  
空间  $O(n)$

Search: 找所有符合的节点  $O(\log n)$

2D: Range Tree: X tree 的 M 个 Y-tree

X-tree: 中位数分裂 1D-树

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

3. Window query

若干点集, 哪些位于 window 内:

1D: 区间树

2D: 区间树

3D: 区间树

4D: 区间树

5D: 区间树

6D: 区间树

7D: 区间树

8D: 区间树

9D: 区间树

10D: 区间树

11D: 区间树

12D: 区间树

13D: 区间树

14D: 区间树

General: (非空但不相交)

1D: 区间树

2D: 区间树

3D: 区间树

4D: 区间树

5D: 区间树

6D: 区间树

7D: 区间树

8D: 区间树

9D: 区间树

10D: 区间树

11D: 区间树

12D: 区间树

13D: 区间树

14D: 区间树

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$

Y-tree: 每个节点  $x$  的 Y-tree 包含所有  $y$  满足  $x \leq y$  的点  
时间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
空间:  $O(n \log n)$  (每个节点  $O(\log n)$ )  
查找:  $O(\log n)$