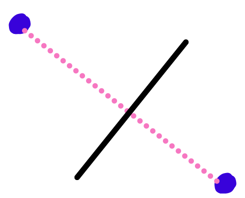
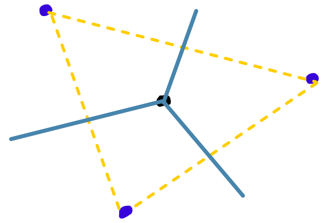
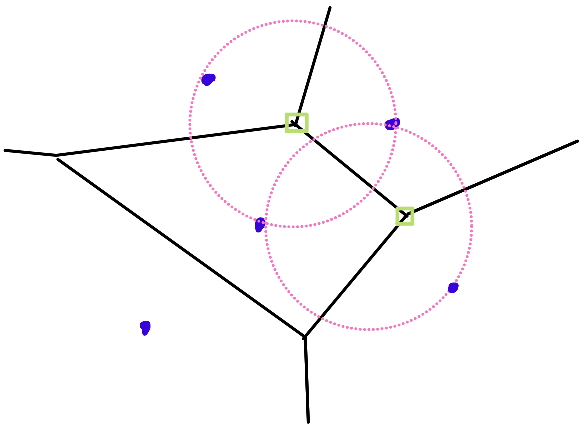
n-sites



Voronoi Diagram

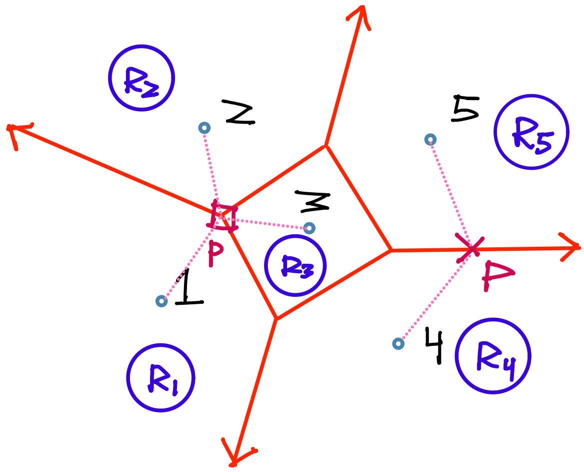


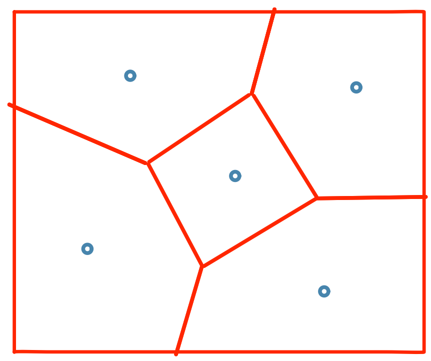


Input:- A set of n point called sites

Output:- A subdivision of the xy-plane into n regions, namely $R_1, R_2, \dots, R_n$ such that each region R_i contains exactly one site (j-th site) and for each point $p \in R_i$, the ith site is the nearest site. (w.r.t Euclidean distance)

The region R_i is called the Voronoi Region for the site i



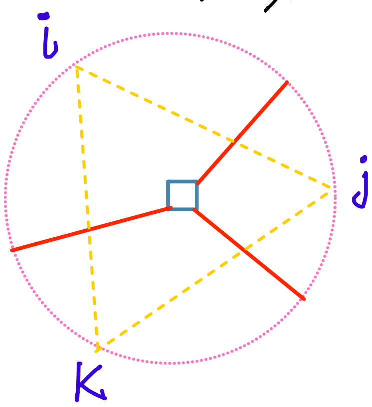


Vertices $= O(n)$
edges $= O(n)$
faces $= O(n)$
(Even in the finite representation of $VD(S)$)

Obs 1:- Any (interior) point on an edge of VD(S) is equidistant from (exactly) two sites

Obs 2:- Any vertex of VD(S) is equidistant from three (or more if four or more sites are concyclic) sites

For simplicity, assume: no four sites are concyclic



$v \in V(VD(S))$
$\Longrightarrow v \in$ circumcenter(i,j,k) for $i,j,k \in S$

Converse of Obs:-

Let i,j,k be any three sites

circumcenter$(i,j,k) \in V(VD(S)) \Rightarrow$ No other site lies in circumcircle(i,j,k) := $C(i,j,k)$

i.e. $C(i,j,k)$ is empty

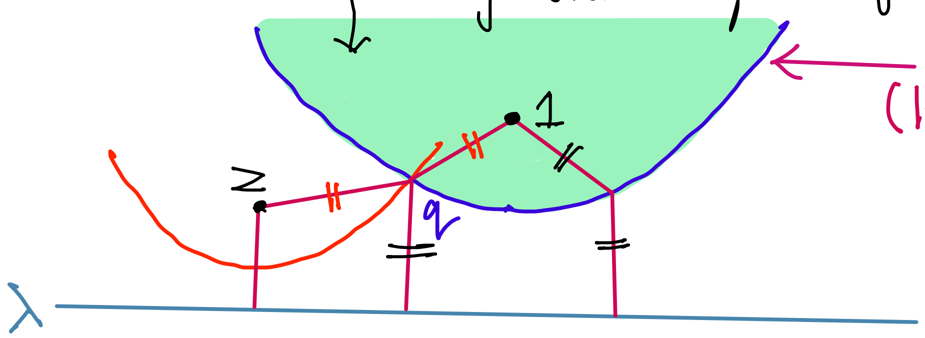
Naive algo:- for any 3 sites $(i,j,k) \longrightarrow \Theta(n^3)$

if $C(i,j,k) = \phi \longrightarrow O(n)$

then $cen(i,j,k) \in V(VD(S))$

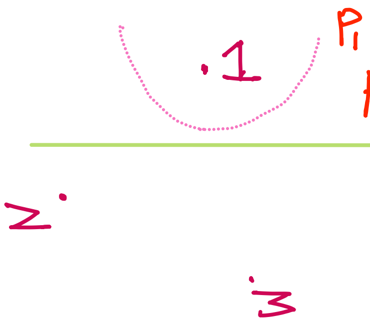
else not

$O(n^4)$ algo



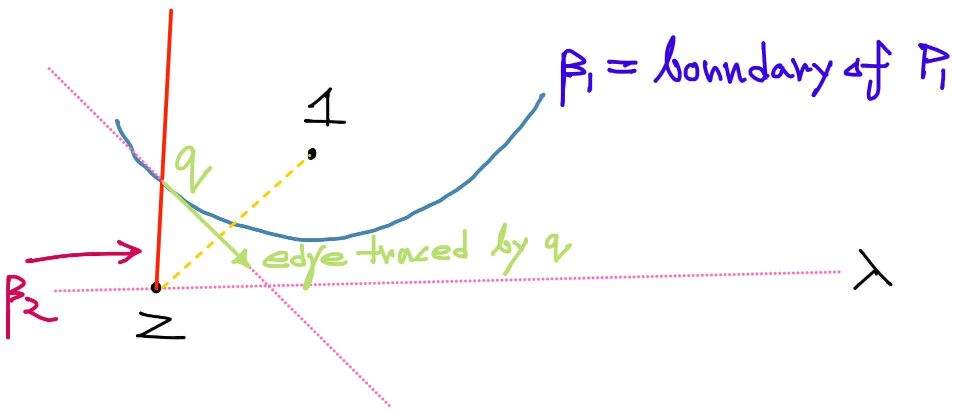
For any point $p \in P_1$ no site below λ can be the nearest site

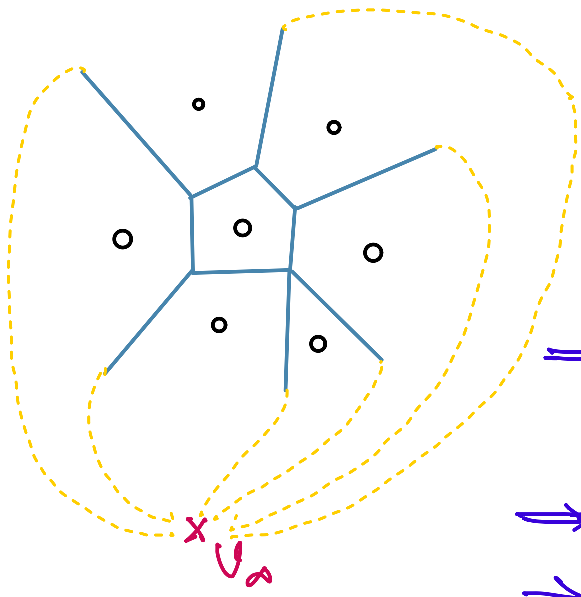
q = a point on some edge of VD(S)



Parabolic region $P_1 \subseteq R_1$

q traces the edge.
(defined by perpendicular bisector of 1 & 2 of) $VD(S)$





G = plannar graph $= (V_G, E_G, F_G)$

Euler formula for plannar graph (connected)

$$\#V_G - \#E_G + \#F_G = 2$$

$$\Rightarrow \#V + 1 - \#E + \#F = 2 \quad \text{--- (1)}$$

$$[\because \#V_G = V+1, \#E_G = E, \#F_G = F]$$

$$\Rightarrow \#V - \#E + n = 1 \quad \text{--- (1A)}$$

$$\Rightarrow \#V = \#E - n + 1$$

$$\sum_{v \in V_G} deg(v) = 2\#E_G = 2\#E \quad \text{--- (2)}$$

for every vertex $v \in V_G$, $deg(v) \geq 3$

$$\Rightarrow \sum_{v \in V_G} deg(v) \geq 3 \cdot \#V_G = 3(\#V+1) \quad \text{--- (3)}$$

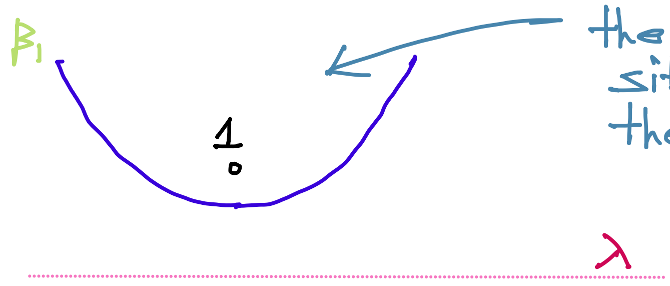
$$(2) \,\&\, (3) \Rightarrow 2\#E \geq 3\#V + 3 \quad \text{--- (4)}$$

$$(1A) \,\&\, (4) \Rightarrow 2\#E \geq 3\#E - 3n + 6 \Rightarrow \#E \leq 3n - 6 \quad \text{--- (5)}$$

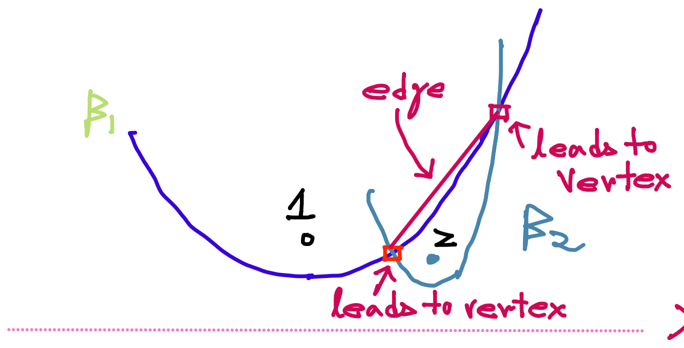
$$\Rightarrow \#E = O(n)$$

(1A) & (5) ⇒

$$\# V \leq 3n-6-n+1 = 2n-5$$

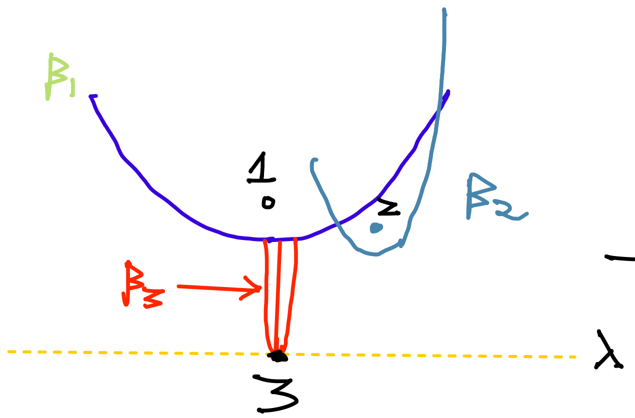


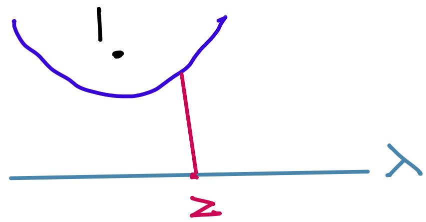
For any point inside β_1, site 1 is the nearest site among all n sites, no matter what and how the sites are located below λ.



For any point inside $\beta_1 \cup \beta_2$, either site 1 or site 2 is the nearest among all given sites irrespective of their locations below λ.

The edge between s_1 and s_2 is discovered (and inserted in the data structure DCEL) when λ encountered β_2.

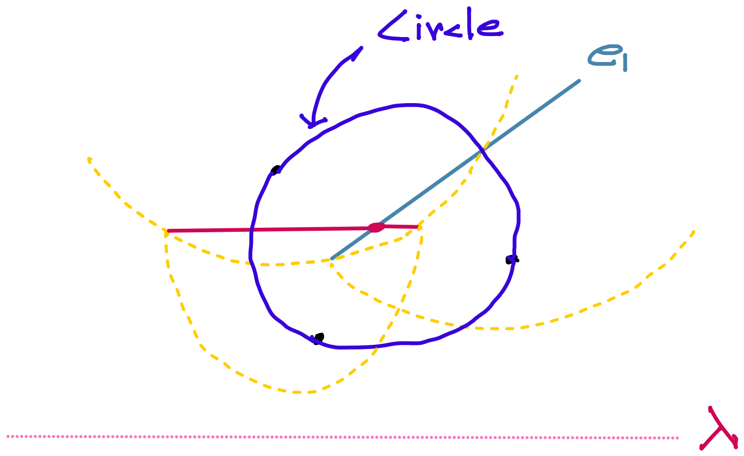


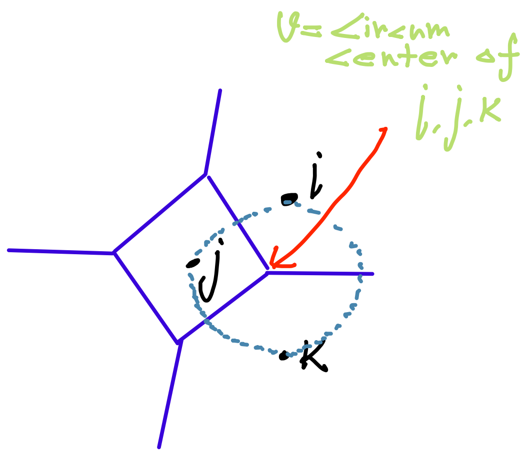


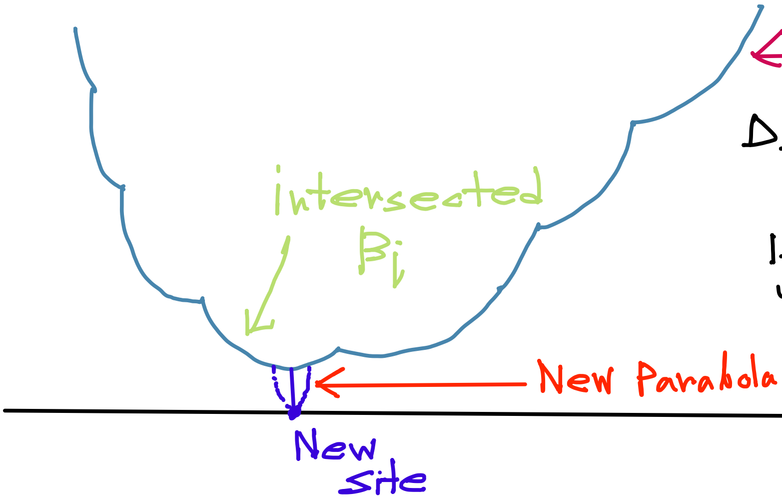
The new parabola β_3 intersects β_1

⇒ A new edge is created in VD(S)

The new edge is given by the perpendicular bisector of $s_1 s_3$







← beach line (β)

DS: left to right ordered parabolas (sites) of β

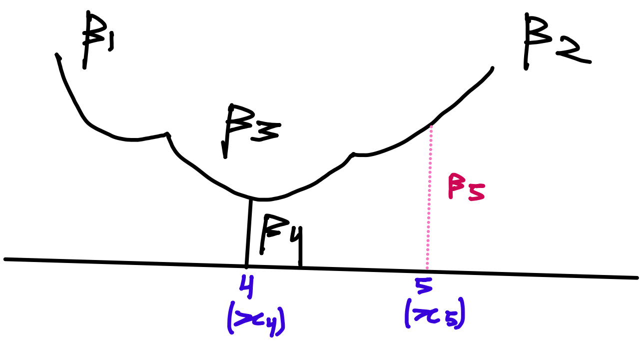
If β has K parabolic arcs, then #sites (possibly repeated) stored in DS(β) = K

λ

Changes in the beach line (β)

1) New parabolic arcs appear
2) Existing parab arcs disappear

Only when λ encounters when a new site



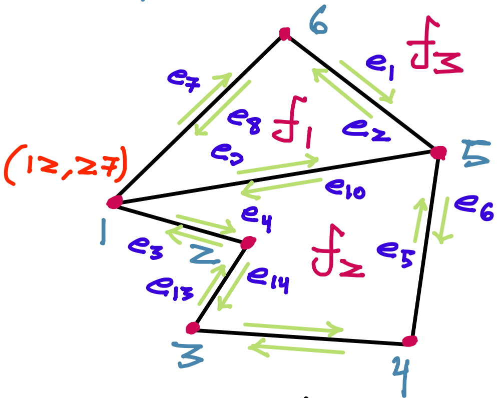
$\beta_3 \rightarrow \beta_3$(left) & β_3(right)

β_4 is new addition

β_3 will be deleted from T_β

β_3(left), β_4(right), β_3(right) will be inserted in T_β in left-to-right order

Doubly Connected Edge list (DCEL) = (V, E, F)



V = {Vertices}

E = {Edges}

F = {Faces}

For every face $f_i \in F$, we consider a fixed ordering of its vertices (say, anticlockwise ordering)

Interior of each f_i lies to the left of each edge of f_i.

$f_1 = \{6,1,5\}$ Start Vertex is not impt.

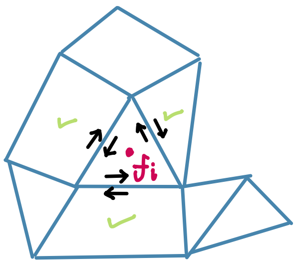
$f_2 = \{4,5,1,2,3\}$

$f_3 = \{3,2,1,6,5,4\}$

Twin Edges:- For every undirected edge e, we have two oppositely-directed edges in E. These are called twin Edges.

A Query Example:-

Given a face f_i, find all faces f_j that are adjacent (i.e share a common undirected edge) to f_i



Edge List (E)

edge ID	Source	twin	face ID	next edge	prev edge
1	6	2	3		
2	5	1	1		
3	2	4	3		
4	1	3	2		
⋮					
9	1	10	1	2	

V

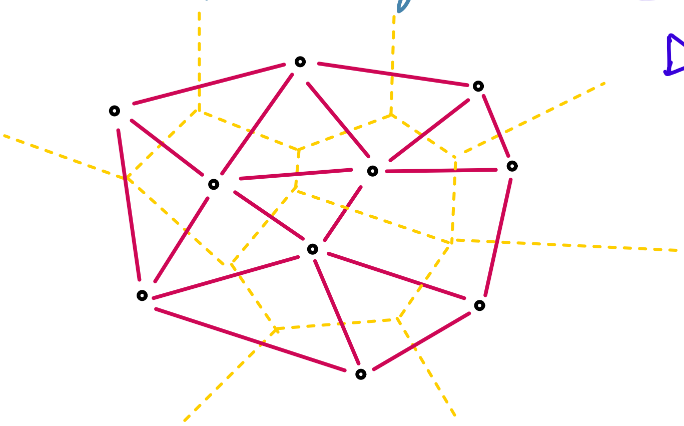
ID	x	y	e_{out} (any one)
1	12	27	9 or 7 or 4

F

ID	incident edge (any one)
1	2
2	5
3	11

Q) Prepare a collection of n linked lists (n = # Vertices) in which each linked list $L(v)$ corresponds to a unique Vertex V and $L(v) = \{$all outgoing edges from $v\}$ — $O(n)$ time

Delanuay Triangulation



S = set of n points

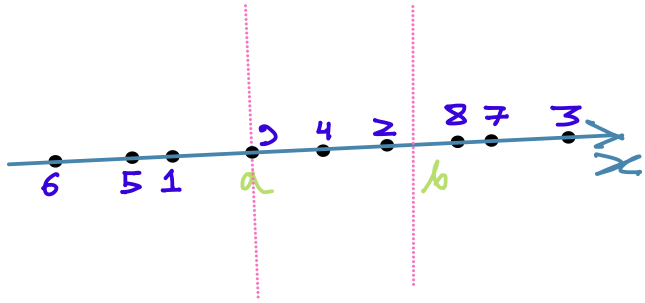
DT(s) = set of non-intersecting triangles with S as the vertex set.

DT(s) is the dual of VD(s)

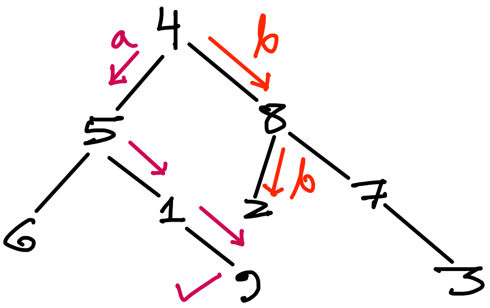
One Dimensional Range Tree

Input:-
- A set of n real values: $S = \{x_1, x_2, \dots, x_n\}$
- An interval $[a, b]$

o/p:- All points in $\underbrace{S \cap [a,b]}_{\#K}$



Ht-balanced BST



Notations

T = range tree with n elements: $x_1, \dots, x_n$
(# nodes > n)

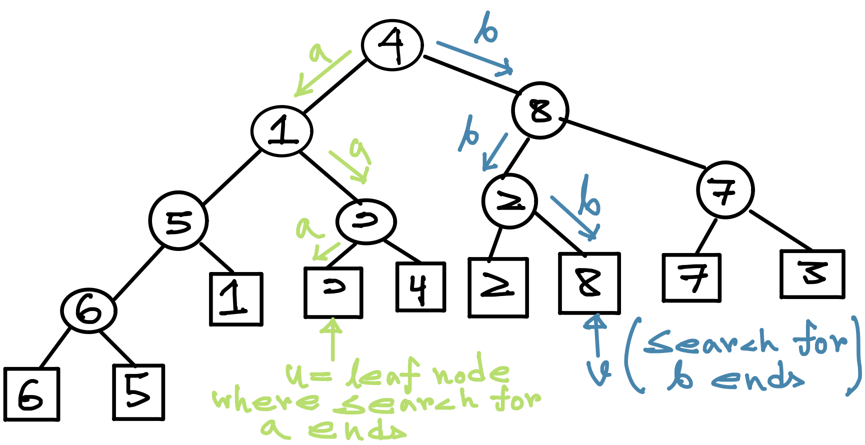
$T(v)$ = subtree rooted at node v of T.

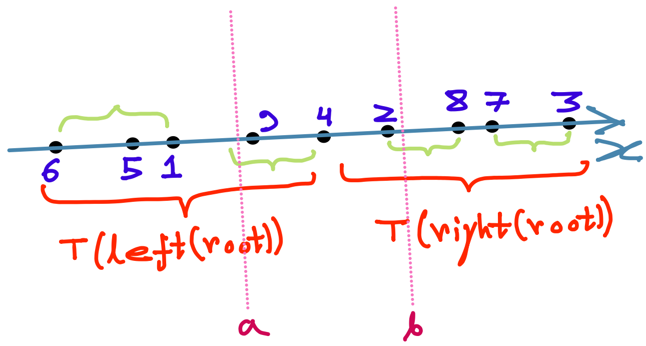
$left(v)$ = left child node of v.

$right(v)$ = right child node of v.

x_v = value stored at v.

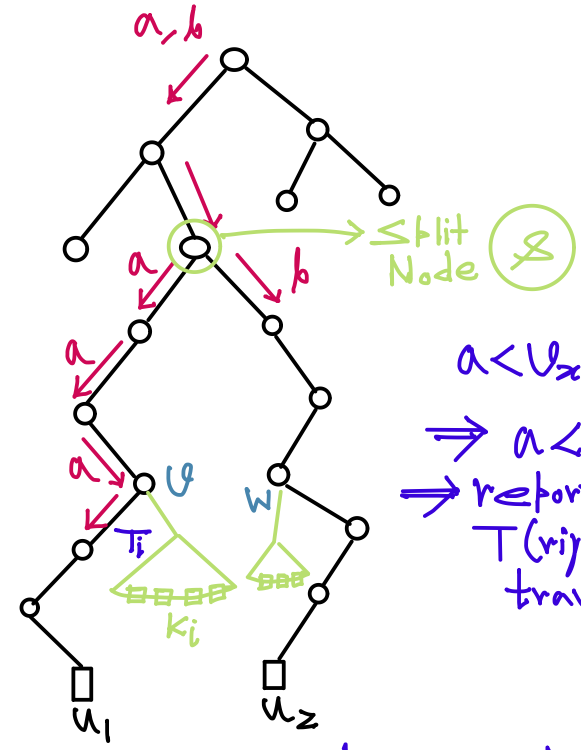
- T has n leaf nodes containing $x_1, \dots, x_n$
- All internal nodes (i.e non-leaf nodes, including the root) are used to aid the searching
- For any internal node v:
  - x_v = value of the rightmost leaf node of $T(left(v))$
  - All elements of $T(left(v))$ < all elements of $T(right(v))$
  - $T(v)$ is balanced.





Nodes in T = O(n) [Prove it using the fact that T is ht-balanced]

Query Q :-
[a, b]



$a < U_x < b$

$\Rightarrow a < T(\text{right}(u)) < b$

$\Rightarrow$ report all leaf nodes in $T(\text{right}(u))$ whenever we traverse left of u

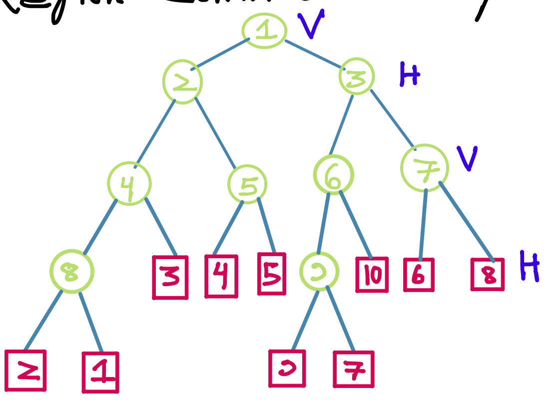
Similar argument holds for a node w on the search path from s to u_2. If we traverse right from W, then all leaf nodes in $T(\text{left}(W))$ are in the sol^n.

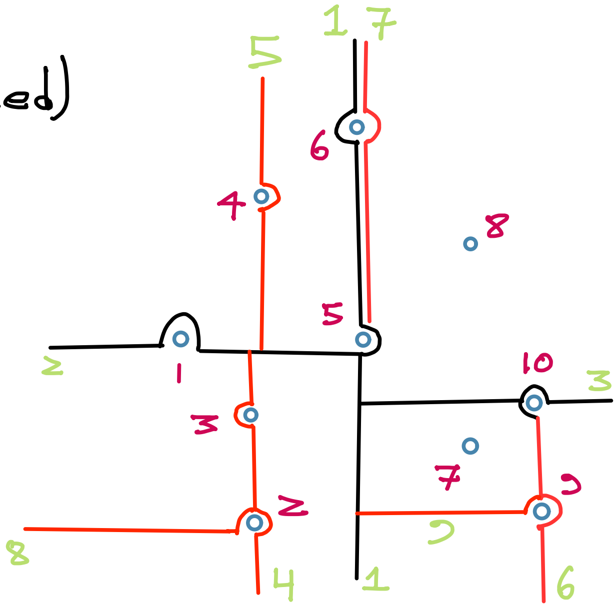
Time $s \to u_1 = O(\log n)$ $\quad s \to u_2 = O(\log n)$

Reporting Time $= O(k)$

Kd-tree

- Multidimensional data structure
- Static Data Structure — admits efficient query
- Can be constructed efficently
- Can be extended to dynamic d.s
- Balanced binary tree (height-balanced)
- Each Region Contains Exactly one point.





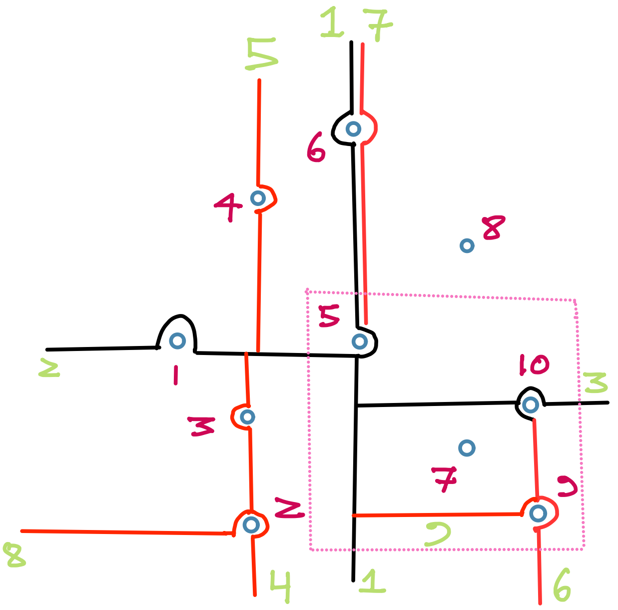
→ Internal node — Stores splitting line

→ Leaf node — Stores a point.

Construction :-

$$T(n) = 2T\left(\frac{n}{2}\right) + O(n) \Longrightarrow T(n) = O(n \log n)$$

Query :- Given a kd-tree T and a query Q = axis-parallel rectangle, find the points stored in T that are contained in Q.



Kd-Tree = T

Query Processing

root of $T = r$

node of $T = v$

left child of v = left(v)

right child of v = right(v)

# of nodes = n

$R(v)$ = Region corresponding to the node v

$T(v)$ = Subtree rooted at v

for each node $x \in T(v) - v$

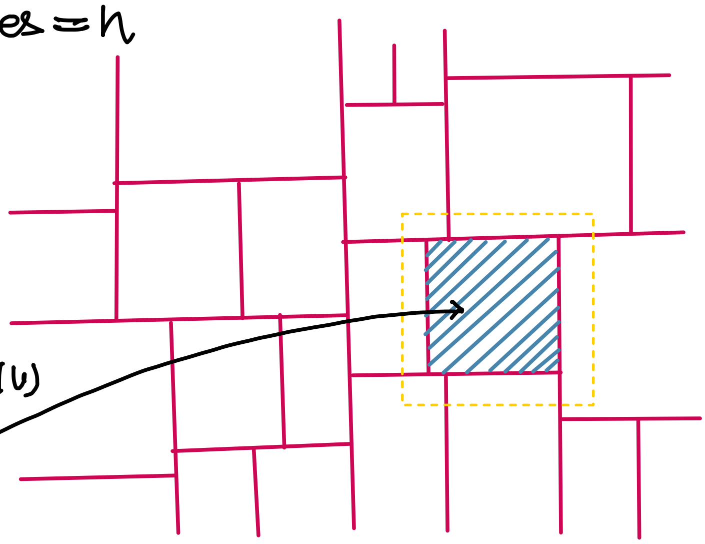
$R(x) \subset R(v)$

$$\bigcup_{x \in V(T(v)) - v} R(x) = R(v)$$

Recursion

$R(v)$'s sides are obtained from the ancestors of v.

$\Longrightarrow$ Regions are obtained as we traverse down.



Traverse $T(v)$ and report its leaf nodes

Two possible cases

- $R(v) \subseteq Q$ — Call REPORTPOINTS(v)
- $R(v) \cap Q \neq \phi$ — TRAVERSE(left(v)) and TRAVERSE(right(v))
- $R(v) \cap Q = \phi$ — Don't traverse

$R(v) \cap Q \neq \phi \Leftrightarrow$ a side of Q intersects $R(v)$.

$\Longrightarrow$ Asymptotic upper bound on the number of regions intersected by Q = asymptotic upper bound of the number of regions intersected by the $\underline{\underline{\text{four sides}}}$ of Q.

$\quad\quad$ $\|$ One side

Q:- What is the asymptotic upper bound of #regions intersected by a vertical line?

Let $f(n)$ = max # regions intersected by any vertical line l.

$$f(n) \leq \underset{\text{level 0 (root)}}{1} + \underset{\text{level 1 (left(v) or right(v))}}{1} + f\left(\frac{n}{2}\right) \longleftarrow \text{Wrong}$$

[$\because f(n)$ is defined at the root level where spliting line is | but $f\left(\frac{n}{2}\right)$ is at the next level where sp. line is —]

$$f(n) = 2 \text{ if } n = 1, 2$$
$$= 2 + 2f\left(\frac{n}{4}\right)$$

$$f\left(\frac{n}{4}\right) = f\left(\frac{n}{2^2}\right) = 2 + 2f\left(\frac{n}{2^4}\right) = 2 + 2\left(2 + 2f\left(\frac{n}{2^4}\right)\right)$$

$$f\left(\frac{n}{2^4}\right) = 2 + 2f\left(\frac{n}{2^6}\right)$$

$$f\left(\frac{n}{2^6}\right) = 2 + 2f\left(\frac{n}{2^8}\right)$$

$$= 2 + 2\left(2 + 2\left(2 + 2f\left(\frac{n}{2^6}\right)\right)\right)$$

$$= 2 + 2\left(2 + 2\left(2 + 2\left(2 + 2f\left(\frac{n}{2^{2\times 4}}\right)\right)\right)\right)$$

$$= 2 + 2^2 + 2^3 + 2^4 + 2^4 f\left(\frac{n}{2^{2\times 4}}\right)$$

Let $n = 2^t$

$$= 2 + 2^2 + 2^3 + \dots + 2^s f\left(\frac{2^t}{2^{2s}}\right)$$

$t = 2s$ or $2s + 1$

$t = 2s$: $f(n) = 2^s - 1 + 2^{s+1}$
$= 3 \cdot 2^s - 1$
$= 3 \cdot 2^{t/2} - 1$
$= 3\sqrt{n} - 1$

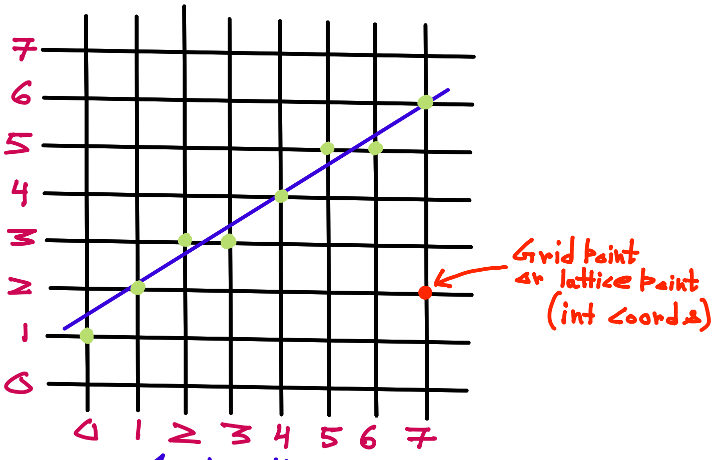
$t = 2s + 1$:
$f(n) = 2^s + 1 - 2^{s+1}$
$= 3 \cdot 2^s - 1$
$= 3 \cdot 2^{\frac{t-1}{2}} - 1 = \frac{3}{\sqrt{2}}\sqrt{n} - 1$

Total Query time $= O(\sqrt{n}+k)$ $k = \#$ of points in Q

Orthogonal Range Query

1) 1D range tree
2) Kd-tree
3) Higher Dimensional Range Tree. (Self-Study)

Geometry of Numbers:-



Grid Lattice

double x=3.1; → Not exactly stored in Computer

$3+\frac{1}{10}$

double x=3.5;

$3+\frac{5}{10} = 1+\frac{1}{2} = 11.1_2$

$\frac{a}{b} = \frac{c_1}{2} + \frac{c_2}{2^2} + \frac{c_3}{2^3} + \cdots + \frac{c_n}{2^n} \Leftrightarrow \frac{a}{b}\Big|_{10}$ has an exact binary representation

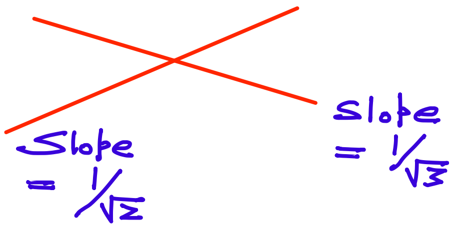
$(a<b)$ $c_j = 0$ or 1 for $j = 1\ldots n$

$\Updownarrow$

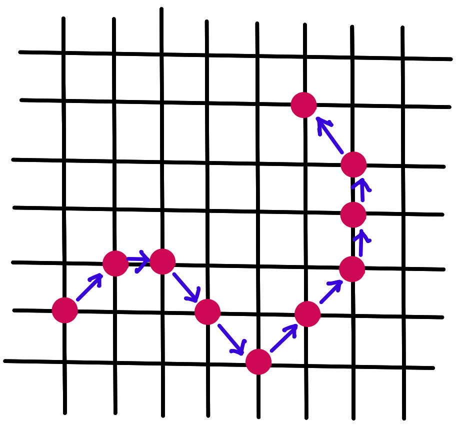
$\frac{a}{2^p}, p \geq 1$

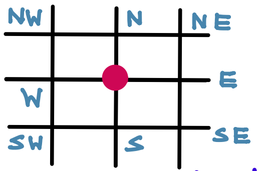
float $y = 3.625 = 3 + \left(\frac{1}{2}+\frac{1}{2^3}\right)$ 11.101000

$= 0.625$

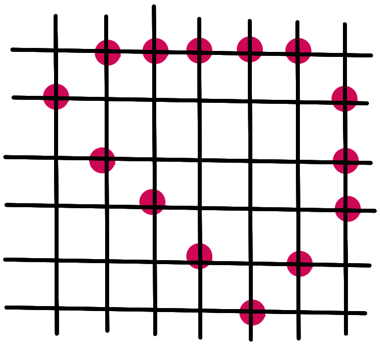


Digital Curve


Open D.C



8-Neighbourhood (8N)

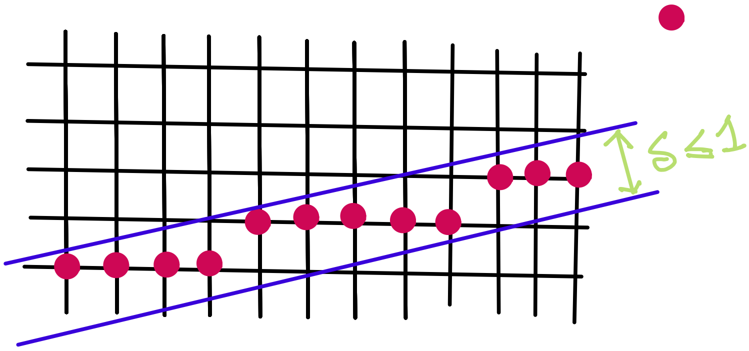

Closed D.C

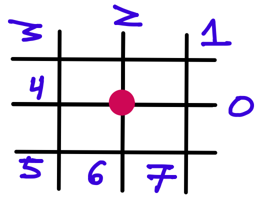
Given:- S = Sequence of n pixels $\{p_1, \ldots, p_n\}$

To decide:- $\exists$? real line L s.t $d(p_i, L) \leq 1/2$ $\forall i$

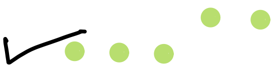
If yes, then S is called a digitally straight segment (DSS)

Prob:- Given S, decide whether S is a DSS.

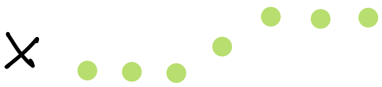




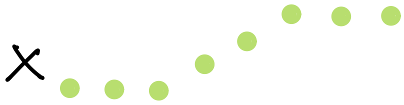
Chain Codes = 0, 1, ..., 7



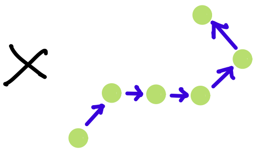
✓ 0010



× 001100



× 0011100



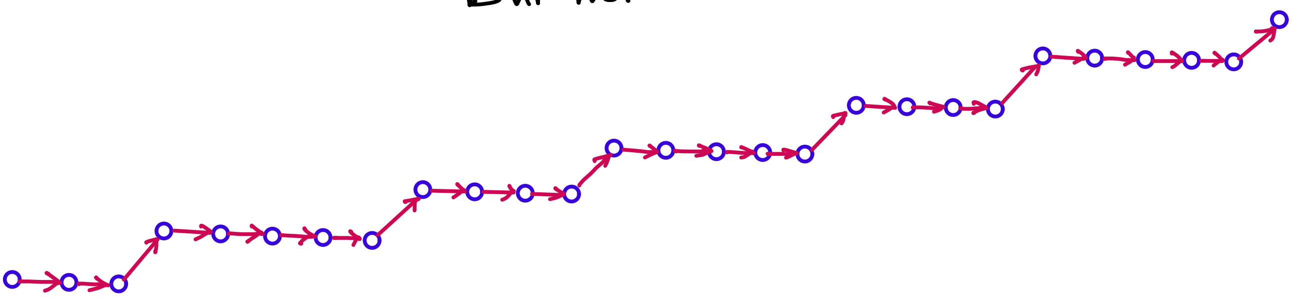
× 10013

	DSS?
000100001000100 (Singular ↓ at the first 1; Non-Singular ↓ over 000)	Yes
000100001210000	No
000100011001	No
0101010	Yes

(S_1) At most two chain codes will be present and they will differ by at most 1 (mod 8). One of them will occur singly (singular element). And the other one is non-singular

001000100000100 $S_1 \Rightarrow$ Yes

But not DSS



Start

Chain Code Sequence:-

$001\overbrace{0000}^{4}1\overbrace{000}^{3}1\overbrace{0000}^{4}1\overbrace{000}^{3}1\overbrace{0000}^{4}1 \longrightarrow$ Satisfies (S_1)

$001\underbrace{00000}_{5}1\underbrace{00}_{2}1\underbrace{0000}_{4}1\underbrace{000}_{3}1\underbrace{0000}_{4}1$

s and n should be distributed as much uniform as possible in order that the sequence is digitally straight. (S_1 does not capture this concept).

(S_2) The run-lengths of n differs by atmost 1.

Ex:- Run Length Sequence

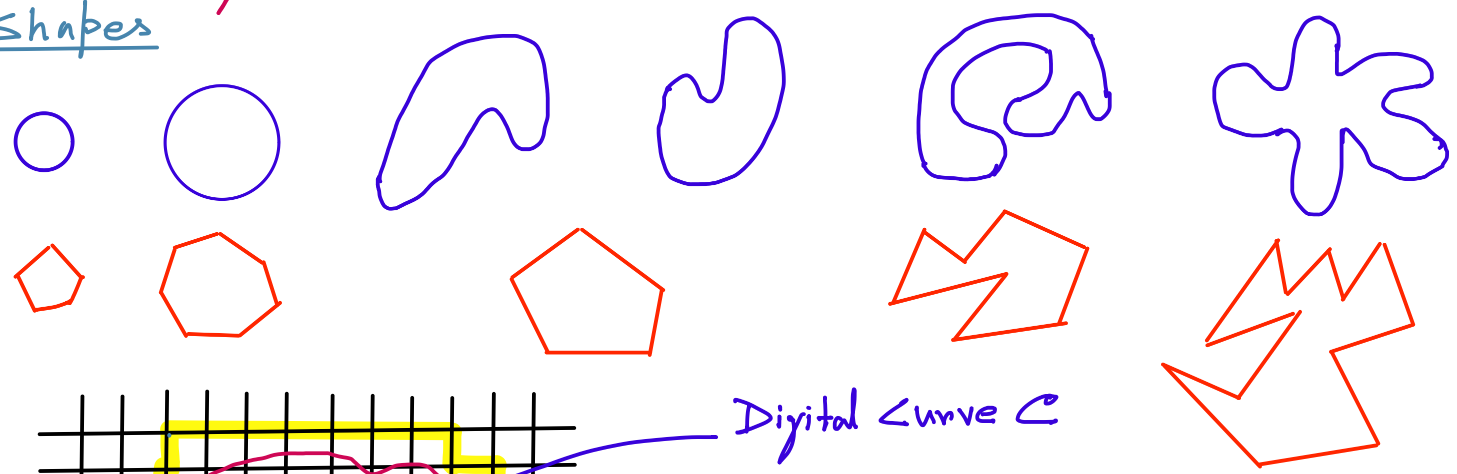
43434 ✓
52434 ×
44334 ✓ ⟷ S_3 fails

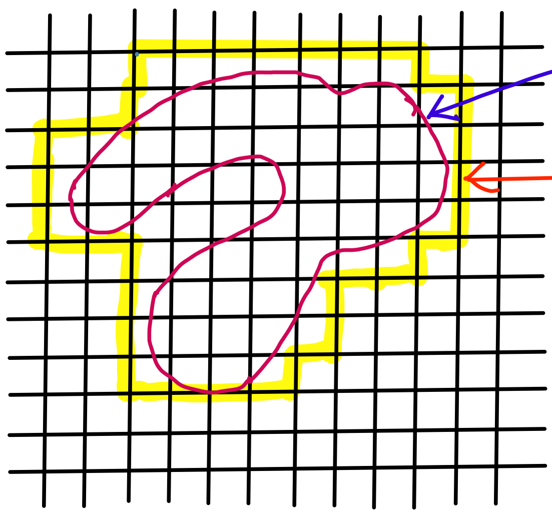
(S_3) one run-length will be singular

$4\,3\,\underbrace{4\,4\,4}_{3}\,3\,\underbrace{4}_{1}\,3\,4$ ⟵ S_3 true

(S_4) Recursively, the non-singular run-lengths of run-lengths should differ by at most 1.

Shapes

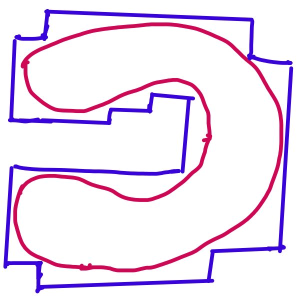




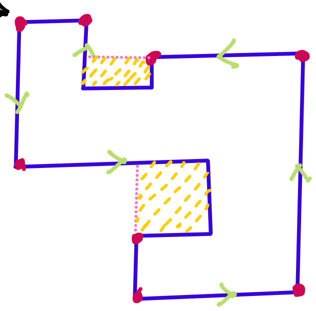
Digital curve C

Isothetic cover of C = Grid Polygon of smallest size whose interior contains C.

Ex: rectilinear convex polygon
orthogonally convex polygon



Start →

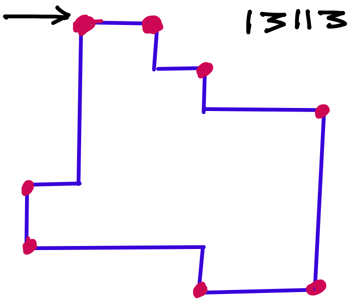


Orthogonal Polygon but not ortho-convex

• = 90° = 1
the unlabeled = 270° = 3

$11\,\underbrace{33}\,11111\,\underbrace{33}\,1$

S →



131131113131