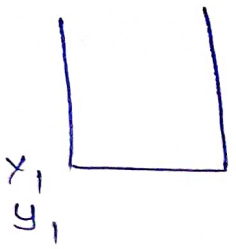
Priority Search Tree

1. Hybrid of Priority Q. + BST.
2. Tree data st. to store points in 2D.
3. Extension of Priority Q + Improve search time.
4. Set of 2D points ordered by priority + at the same time split by x / y coordinates.
5. Essentialy a heap + BST.
6. Each Node - point in data set. + key value (divides the points in left sub tree & right sub tree).
7. Priority - can be either "x" / "y" - Not both.
8. Used for unbounded search $[x_1, x_2]$ $[y_1 \ldots$
9. Unbounded Range search.

Ex.



Construction Process.

* let S be the set of Points.

(1) If S is empty return Null.

(2) find point P (with highest value of Y coor.) from S and set P as the root R

(3) find S-P (remove P from S)

(4) If S-P is empty, return P.

(5) find x(P) such that x divides points in S-P into 2 equal halves (sets) on x coordi.

(6) Reursively apply above steps on left & rt. child.

Ex. Given Points

(25,35)
(50,10)
(85,15)
(90,5)
(5,45)
(35,40)
(60,75)
(80,65)

① Sort on x axis.

(5,45)
(25,35)
(35,40)
(50,10)
(60,75)
(80,65)
(85,15)
(90,5)

S = (5,45)(25,35)(35,40)(50,10)(60,75)
(80,65)(85,15)(90,5)

P = (60,75) Highest y co-ordi.

S-P = (5,45)(25,35)(35,40) | (50,10)(80,65)
(90,5).

x(P) = 43 (Median).

↓ create PST.

S = (50,10)(80,15
(90,5)

P = (80,15)

S-P = (50,10) | 90

create PST.

S = (5,45)(25,35)(35,40)

~~P~~ P = ~~(60,~~ (5,45)

~~SP~~ ⇒ S-P = (25,35) | (35,40)

x(P) = 30 (Median).

$S = (25, 35)$

$P = (25, 35)$ A

$S - P =$ Empty

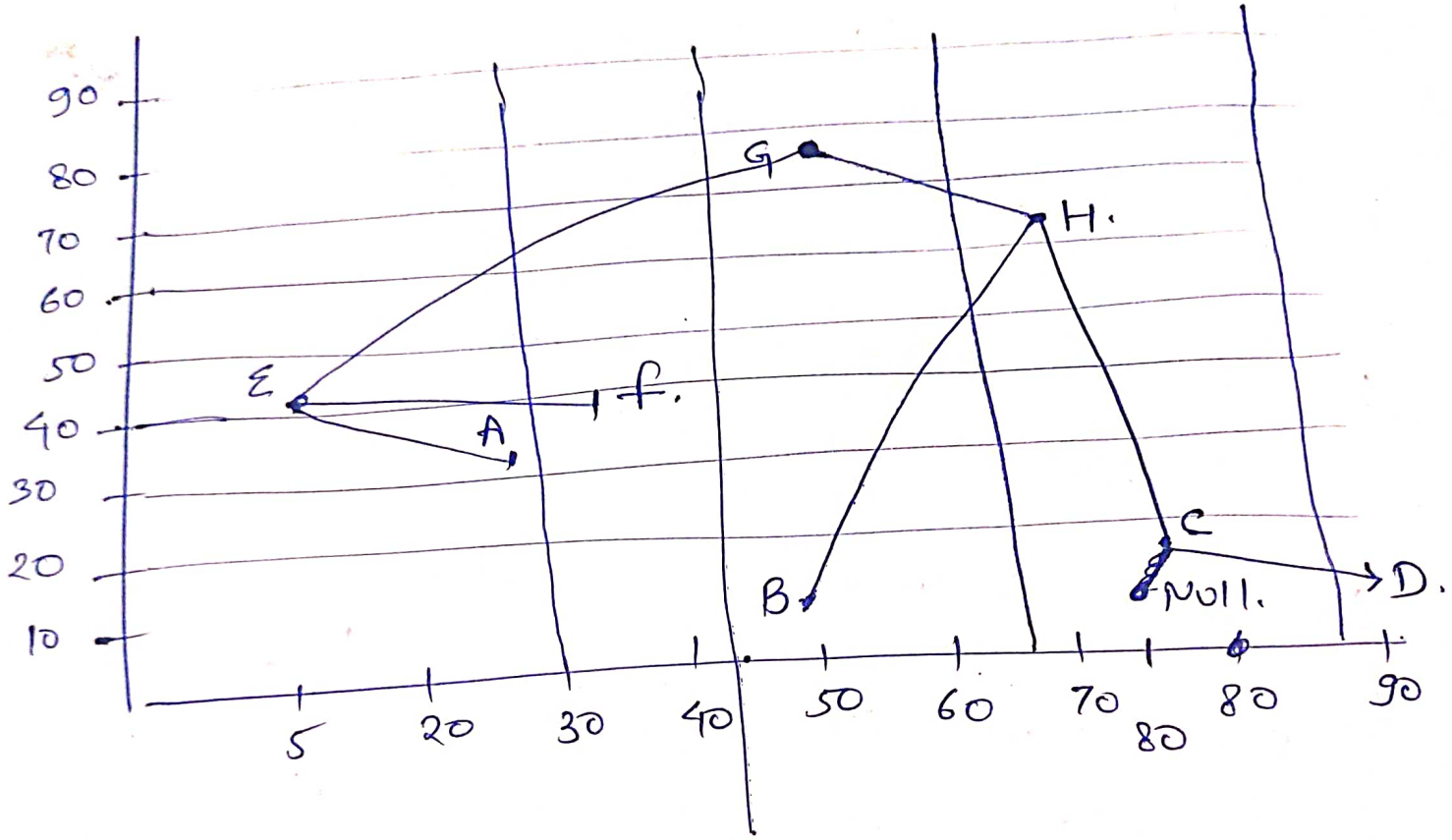
return (A)

$S = (35, 40)$

$P = f\ (35, 40)$

$S - P =$ Empty.

return f.



43.

pdf.

Brown Uni. Computer Science.

P.S.T. - Part I