Homework #4

Trees

Anastasiia Konoplina

---

2. Describe the deletion procedure from red-black tree. What are the possible different cases and how can we ensure re-balancing. How many operations (recolorings, rotations) is needed?

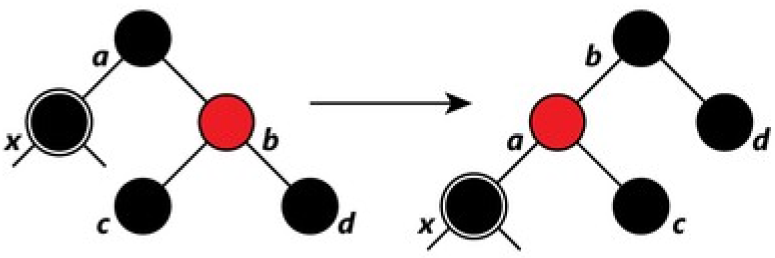
In general deletion procedure of some node in black-red tree can be divided into 2 steps:

1) Deletion:

- If a node does not have children, no additional actions (# of operations = 1)
- If a node has 1 child – move this child up on a place of deleted node (# of operations = 2)
- If a node has 2 children. New node which replaces deleted one is a node with "next value element". It means we go to right child and then to left until left child exists (the "most left" child in right subtree) (# of operations = 2)

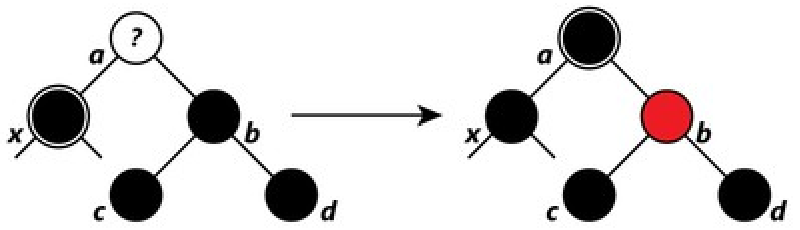
2) Ensure re-balancing. Deletion of red node does not affect tree's "parameters". But in case of black node deletion there are several cases:

1. If sibling is red → rotation between parent and sibling (so that sibling becomes parent). Then perform recoloring. (# of operations = 2)

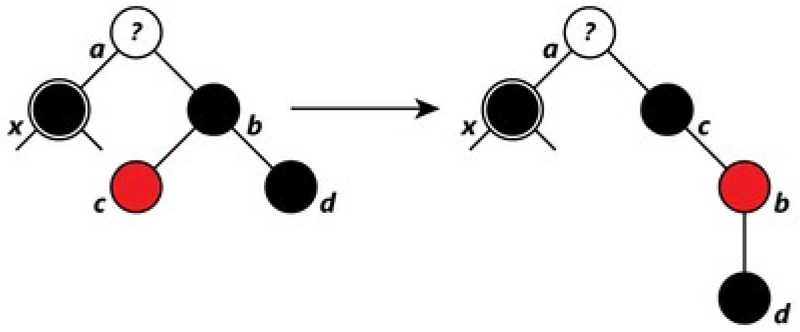


2. If sibling is black there are 3 cases:

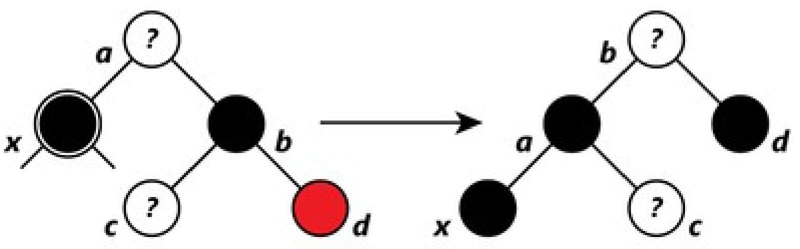
   - If both sibling's children are black. Recolor sibling to red, its parent – to black. (# of operations = 1)



   - If sibling's right child is black and left is red. In this case we need to recolor sibling and left child and perform rotation. (# of operations = 2)



   - If sibling's right child is red. Recolor sibling to parent's color, child and parent – to black. Perform rotation (# of operations = 2)
   -



3. In the lecture we said rotations can not be used for k-d trees. Explain why. How might one try to achieve rebalancing of dynamic k-d trees?

Rotations cannot be used for k-d trees since in for example binary trees we insert elements one by one, but for building k-d trees usually whole dataset is used, so that in k-d trees different “axes” compare different “aspects” of node, and if we rotate the tree for balancing purpose we would break its structure.

I think that re-building of some part of kd-tree. For example divide the tree into several sub-trees and rebuild some of these subtrees id needed.

4. Define the random projections based search tree for 2-D data. First, define the orthogonal plane between any two randomly chosen points. Next, define formula for deciding whether a particular point should be in the left or right subtree from the hyperplane. Illustrate this with some 12-15 data points graphically. Draw the points and the full search tree. Provide the formulas needed for traversing the tree for identification of a particular point in the tree. (take note of task 5 when designing the illustration).

Random projection tree

1 (2; 3)
2 (1; 4)
3 (7; 9)
4 (8; 5)
5 (6; 2)
6 (4; 4)
7 (7; 3)
8 (5; 9)
9 (6; 8)
10 (1; 7)
11 (3; 1)
12 (3; 6)

1) Pick 2 random points: 3(7; 9) and 11(3; 1)

2) Define splitting hyperplane:

$$\frac{x - x_2}{x_1 - x_2} = \frac{y - y_2}{y_1 - y_2}$$

$$\frac{x - 3}{7 - 3} = \frac{y - 1}{9 - 1}$$

$8(x-3) = 4(y-1)$

$8x - 4y - 20 = 0$

$2x - y - 5 = 0$ – formula of line by these 2 points

$C(\frac{7+3}{2}; \frac{9+1}{2}) \Rightarrow C(5; 5)$ – middle point

$2(y - y_1) + 1(x - x_1) = 0$

$2(y-5) + x - 5 = 0$

$x + 2y - 15 = 0$ – splitting hyperlane

3) Define to wich subtree each point belongs:

1. $2 + 2 \cdot 3 - 15 = -8 \Rightarrow$ left
2. $1 + 2 \cdot 4 - 15 = -6 \Rightarrow$ left
3. $7 + 2 \cdot 9 - 15 = 10 \Rightarrow$ right
4. $8 + 2 \cdot 5 - 15 = 3 \Rightarrow$ rigt
5. $6 + 2 \cdot 2 - 15 = -5 \Rightarrow$ left
6. $4 + 2 \cdot 4 - 15 = -3 \Rightarrow$ left
7. $7 + 2 \cdot 3 - 15 = -2 \Rightarrow$ left
8. $5 + 2 \cdot 9 - 15 = 8 \Rightarrow$ right
9. $6 + 2 \cdot 8 - 15 = 7 \Rightarrow$ right
10. $1 + 2 \cdot 7 - 15 = 0$ – on line
11. $3 + 2 \cdot 1 - 15 = -10 \Rightarrow$ left
12. $3 + 2 \cdot 6 - 15 = 0$ – on line

Left tree

3. $(7;9)$
4. $(8;5)$
8. $(5;9)$
9. $(6;8)$

1) $3(7;9)$ and $8(5;9)$

2) $\frac{x-x_2}{x_1-x_2}=\frac{y-y_2}{y_1-y_2}$

$\frac{x-5}{7-5}=\frac{y-9}{9-9}$

$0(x-5)=2(y-9)$

$2y-18=0$

$C(6;9)$

$0(y-y_1)+-2(x-x_1)=0$

$-2x+12=0$

$-x+6=0$

3) 3. $-7+6=-1 \Rightarrow$ left
4. $-8+6=-2 \Rightarrow$ left
8. $-5+6=1 \Rightarrow$ right
9. $-6+6=0 \Rightarrow$ on line

---

1) $3(7;9)$ and $4(8;5)$

2) $\frac{x-8}{7-8}=\frac{y-5}{9-5}$

$4(x-8)=-(y-5)$

$4x-32+y-5=0$

$4x+y-37=0$

$C(7,5;7)$

$4(y-7,5)-(x-7)=0$

$4y-30-x+7=0$

$4y-x-23=0$

3) 3. $4\cdot 9-7-23=6 \Rightarrow$ right
4. $4\cdot 5-8-23=-11 \Rightarrow$ left

Right tree

1. $(2;3)$
2. $(1;4)$
5. $(6;2)$
6. $(4;4)$
7. $(7;3)$
11. $(3;1)$

1) $11(3;1)$ and $7(7;3)$

2) $\frac{x-x_2}{x_1-x_2}=\frac{y-y_2}{y_1-y_2}$

$\frac{x-7}{3-7}=\frac{y-3}{1-3}$

$-2(x-7)=-4(y-3)$

$2x-14-4y+12=0$

$2x-4y-2=0$

$x-2y-1=0$

$C(5;2)$

$(y-y_1)+2(x-x_1)=0$

$y-2+2x-10=0$

$2x+y-12=0$

3) 1. $2\cdot 2+3-12=-5 \Rightarrow$ left
2. $1\cdot 2+4-12=-6 \Rightarrow$ left
5. $2\cdot 6+2-12=2 \Rightarrow$ right
6. $4\cdot 2+4-12=0 \Rightarrow$ on line
7. $2\cdot 7+3-12=5 \Rightarrow$ right
11. $2\cdot 3+1-12=-6 \Rightarrow$ left

---

1) $2(1;4)$ and $4(3;1)$

2) $\frac{x-3}{1-3}=\frac{y-1}{4-1} \Rightarrow 3(x-3)=-2(y-1)$
$3x-9+2y-2=0$
$3x+2y-11=0$ $C(2;2,5)$

$3(y-2,5)-2(x-2)=0$

$3y-7,5-2x+4=0$

$2x-3y+3,5=0$

3) 1. $2\cdot 2-3\cdot 3+3,5=-1,5 \Rightarrow$ left
2. $2\cdot 1-3\cdot 4+3,5=-6,5 \Rightarrow$ left
11. $2\cdot 3-3\cdot 1+3,5=6,5 \Rightarrow$ right

---

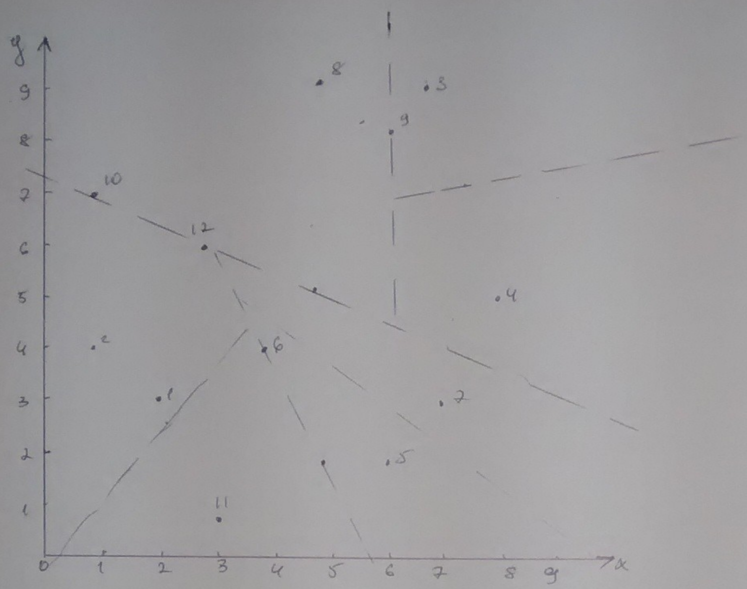
1) ~~5(6;2) and~~ 7(7;3)

2) $\frac{x-7}{6-7}=\frac{y-3}{2-3} \Rightarrow -(x-7)=-(y-3)$ $C(6,5;2,5)$
$-x+y+4=0$

$1(y-y_1)+(x-x_1)=0 \Rightarrow y-6,5-x-2,5=0$

$\boxed{y+x-9}$

3) 5. $6+2-9=-1 \Rightarrow$ left
7. $7+3-9=1 \Rightarrow$ right



Provide the formulas needed for traversing the tree for identification a particular point in the tree.

As I understood we are given some point $(x; y)$ and we need to find it in our tree.

For this we need to take a formula of each hyperplane and depending on gotten result ($>$ or < 0) move to appropriate subtree. Repeat this iteratively untill value is found or does not exist.

*5. Describe the process of searching the **proximity** of a particular point from a random projection tree. E.g. - list all points that are within radius <= sqrt(3) from the search point. Illustrate this on the same example from task 4. Make an example **query** that does not exist exactly in data but that would fetch exactly 4 points within that radius.*

N 5

1) Find distance between given point and other points. Pick points if distance < radius

2) Query: I would use KNN (k nearest neighbor): calculate all distances and pick k elements with the smallest neighbors

1 (2;3)	5 (6;2)	9 (6;8)
2 (1;4)	6 (4;4)	10 (1;7)
3 (7;9)	7 (7;3)	11 (3;1)
4 (8;5)	8 (5;9)	12 (3;6)

Let's take radius = 4 and point 10(1;7)

d(1;10) = 4,1	d(7;10) = 7,2
d(2;10) = 3	d(8;10) = 4,5
d(3;10) = 6,3	d(9;10) = 5,1
d(4;10) = 7,3	d(11;10) = 6,3
d(5;10) = 7,1	d(12;10) = 2,2
d(6;10) = 4,2	

Proximity of 10(1;7) and radius 4:

2(1;4)
12(3;6)

2) Pick 4 closest elements:

~~12(3;6)~~
~~2(1;4)~~
~~1(2;3)~~
~~6(4;4)~~

12 (3;6)
2 (1;4)
1 (2;3)
6 (4;4)

---

6. Define the random projections tree for 3-D and k-D data. Next, define formula for deciding whether a point should be in the left or right subtree from the hyperplane. Try to create an educational example of this situation for 5-dimensional data.

Random projections tree (3-D and k-D)

Splitting hyperplane dimension of k-D tree ~~wt~~ is (k-1)-D. I think the algorithm is the same:

1) Take k points and build (K-1)-D hyperplane.
2) ~~App~~ Calculate the result, if >0 => right subtree, if <0 => left subtree

General formula for (k-1)-D hyperplane:

$$a_1x_1 + a_2x_2 + a_3x_3 + a_4x_4 + \ldots + a_nx_n + a_{n+1} = 0$$

— — — — — — — —

I don't know how to build a hyperplane from n points, but if we have a formula of splitting hyperplane:

$2x_1 + 3x_2 - 4x_3 - x_4 + 2x_5 = 0$ and poin $A(2;3;4;4;4)$ $B(1;1;5,4,2)$

A) $2 \cdot 2 + 3 \cdot 3 - 4 \cdot 4 - 4 + 2 \cdot 4 = 4 + 9 - 8 - 4 + 8 = 9$ => right subtree

B) $2 + 3 - 20 - 4 + 4 = -14$ => left subtree

and so on