Q1:

a)

Insert I, E, A, T, P, R, S, L

15	
14	
13	
12	L
11	
10	
9	I
8	
7	
6	
5	E
4	T
3	S
2	R
1	A
0	P

Now need to insert Y,G. first check 9+3=12, and its occupied by "L". so we check 9+2*3=15, and insert "Y" at index 15. And lastly insert G at 7.

15	Y
14	
13	E
12	L
11	
10	
9	I
8	
7	G
6	
5	E
4	T
3	S
2	R
1	A
0	P

b)

It is easier to make “fail to insert” happen by making the increment number be a factor of table size. In this case, since table size is 16, by making the result given by h_2 to be 8 will create a loop of 2 indices, to be 4 will create a loop of 4 indices, etc. Since ascii letter “O” will make the hash function h_2 return 8, try “O”. first hash function returns index 15. Since Index 15 is occupied by “Y” and index 15 + 8 – 16 = 7 is occupied by “G”, the probe sequence creates a loop between these two indices, and thus “fail to insert” will happen if we try to insert “O”.

c)

i) h_1 has bug.

Then the first element will go to the result of h_1, and second and later will conflict with h_2 and will go to the index returned by h_2. New element cannot be added if h_2 returns a index which it has returned before.

In average, inserting an element requires calling hash function twice.

ii) h_2 has bug.

All elements go to the index returned by h_1. If there is conflict, it goes to index returned by h_2. At most two conflicts can happen in this case. The index returned by h_2 might have been occupied during the first inserting-with-no-conflict stage, and once conflict occurs the table has to be rehashed.

In average, inserting an element requires calling hash function once.

iii) both h_1 and h_2 has bug.

The first element goes to the index returned by h_1, the first element goes to the index returned by h_2(since h_1 index is occupied), and the third element cannot be inserted(since both h_1 and h_2 indices are occupied).

The hash table can only insert two elements (one if h_1 and h_2 returns the same index). An insert will be quickly unavailable.

Q2:

a)

Insert I, E, A, T, P, R, S, L

15	
14	
13	
12	L
11	
10	
9	I
8	
7	
6	
5	E
4	T
3	S
2	R
1	A
0	P

Now need to insert Y. first check index 9(h_1), and it's occupied by "I". so we check index 3(h_2), and it's occupied by "S". Hence we kicks out "I", and place "Y" at index 9(h_1). Then we kicks out "S", and place "I" at index 3(h_2), kicks out "A", and place "S" at index 1(h_2), and finally place "A" at index 11(h_2).

Now need to insert G. simply insert it at index 7.

15	
14	
13	
12	L
11	A
10	
9	Y
8	
7	G
6	
5	E
4	T
3	I
2	R
1	S
0	P

b)

we use the fact that “I” and “Y” have the same returned index for both h_1 and h_2. Since it is not possible to let h_1 return 9 for A-Z, we try to let h_1 return 3 instead, so the candidate is “C” (“S” is already in the table). Since $h_2(C) = 1$, and 1 is already occupied by “S”, “C” stays in 3. Then “I” goes to 9, “Y” goes to 3, “C” goes to 1, “S” goes to 3, “Y” goes to 9…… since four letters “C”, “Y”, “I”, “S” have to stay in 3 indices 1,3,9, which is impossible, “fail to insert” will happen.

c)

1: the letter “P” is neither in index 0 or index 0.(h_1 or h_2)

2: either inserting “L” first or inserting “T” first will make “L” in index 12 and “T” in index 4, not the reverse.

3: R is in index 6, which is returned by h_2, while no letter is occupying index returned by h_1.

Q3:

a)

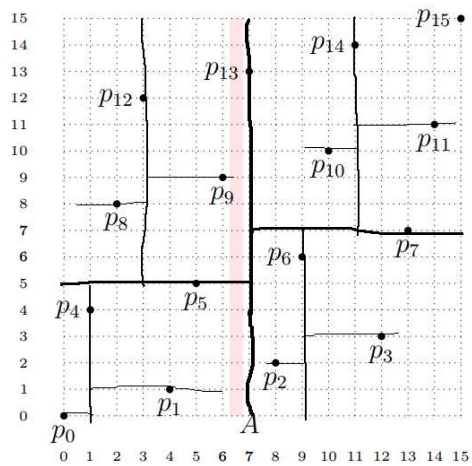
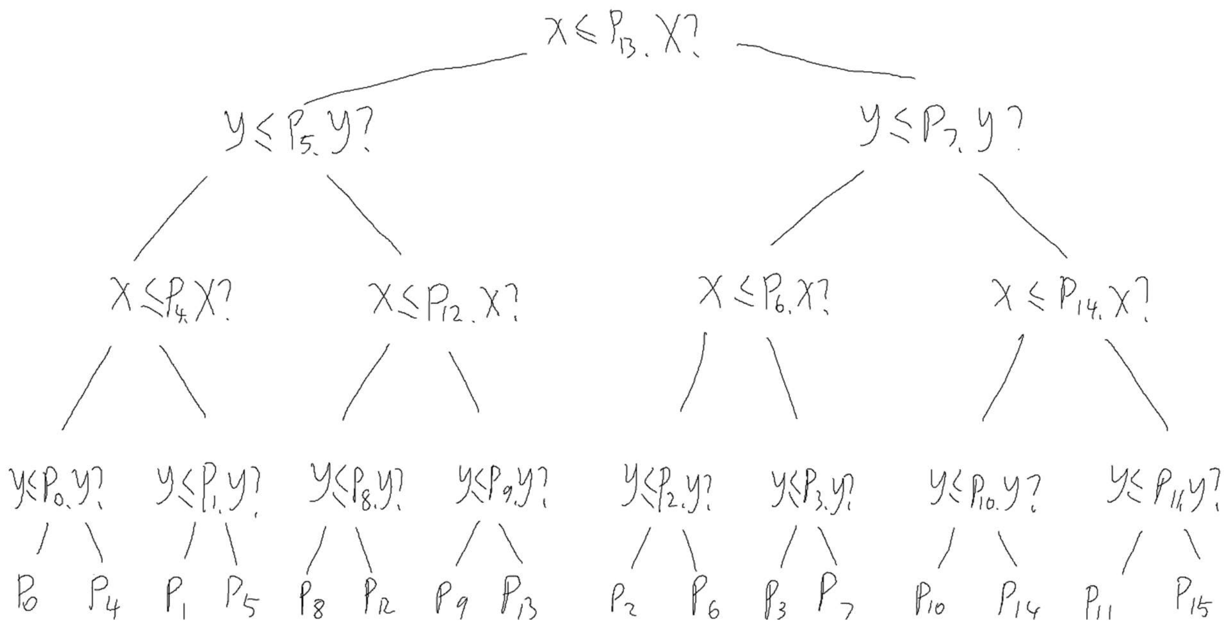
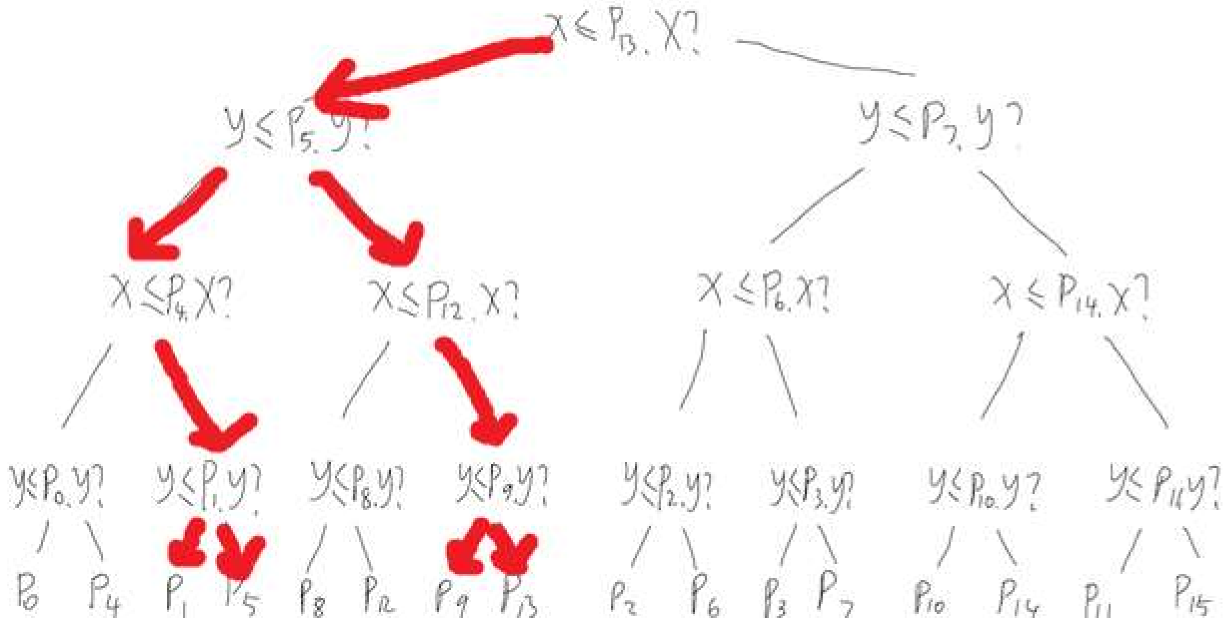


Figure 1: A set of points, and the query-rectangle A.



b)

P_1, P_5, P_9, P_{13} will be reported.



c)

To have such a rectangle to have no point of P but have greatest number of grey nodes, we need to make the rectangle thin and long (as in question b), then half of the time it reports all nodes below and half of the time it reports half of its child nodes.

Given a point, consider its 4 quadrants. Since the rectangle goes through two of it, algorithm must recur on 2 of 4 quadrants. Assume $Q(n)$ represents the number of grey nodes, then

$$Q(n) \geq 2Q\left(\frac{n}{4}\right) + \Omega(1)$$

and this will solve to $Q(n) \in \sqrt{n}$.

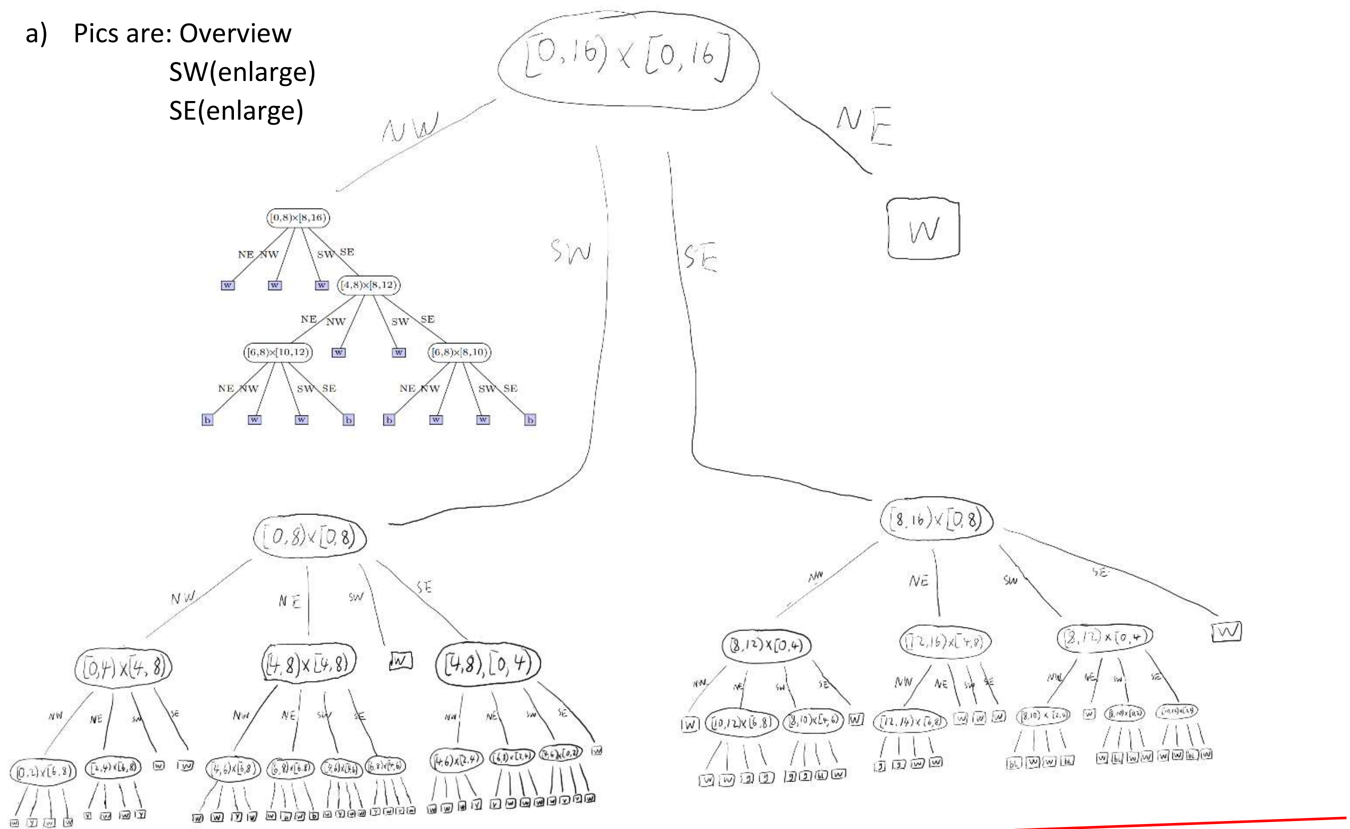
Q4:

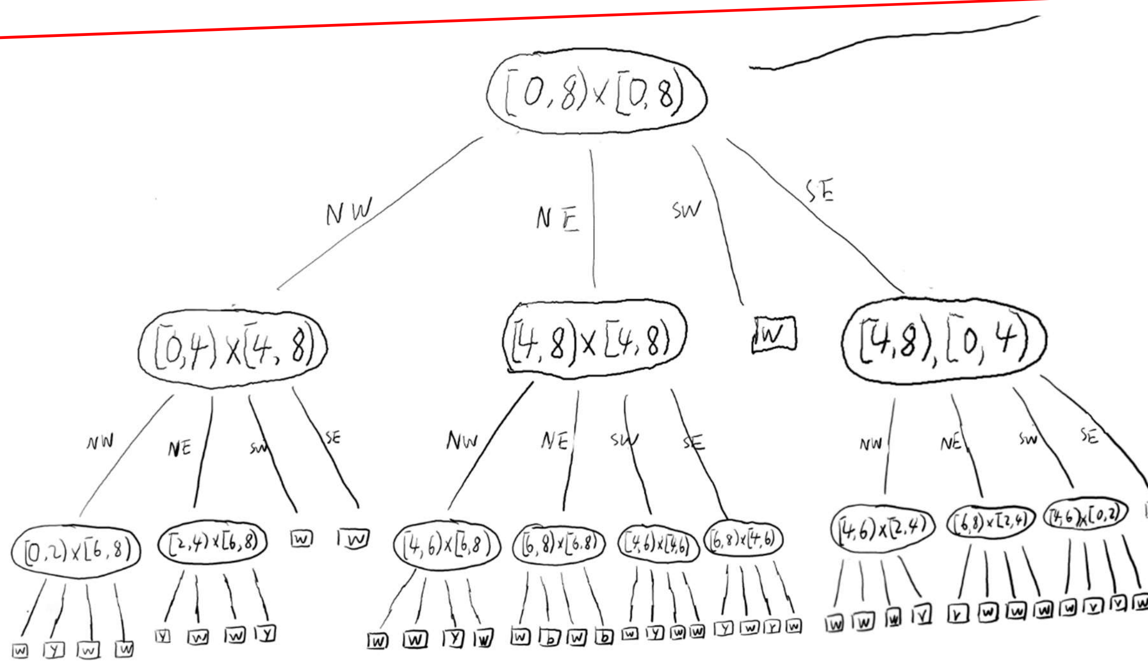
We should use range trees with 4 dimensions.

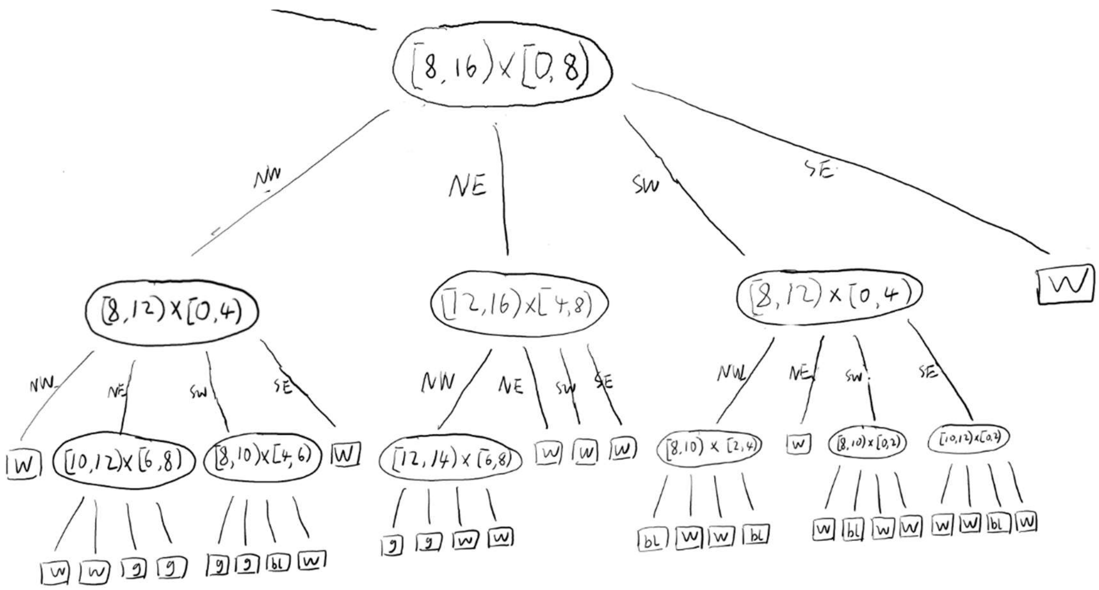
First

Q5:

a) Pics are: Overview
SW(enlarge)
SE(enlarge)







[8,16)×[0,8)
NW
NE
SW
SE
[8,12)×[0,4)
[12,16)×[4,8)
[8,12)×[0,4)
W
[10,12)×[6,8)
[8,10)×[4,6)
[12,14)×[6,8)
[8,10)×[2,4)
[8,10)×[0,2)