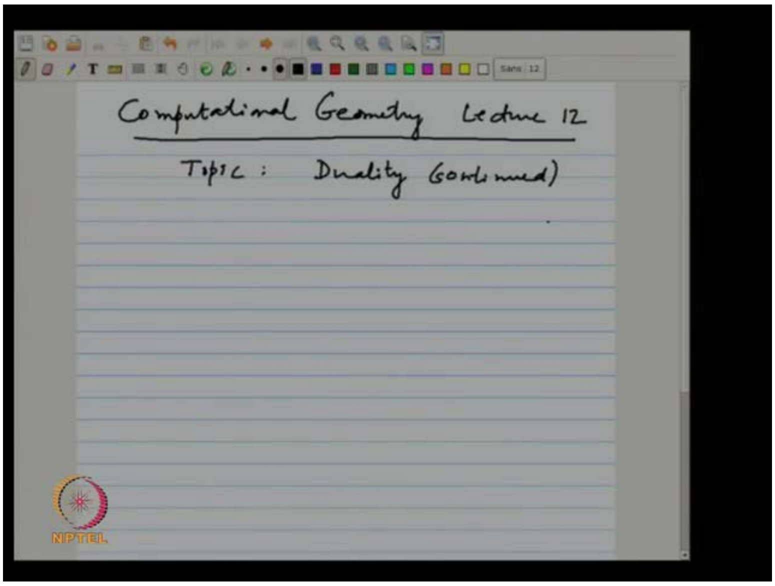
Computational Geometry
Prof. Sandeep Sen
Department of Computer Science & Engineering
Indian Institute Of Technology, Delhi

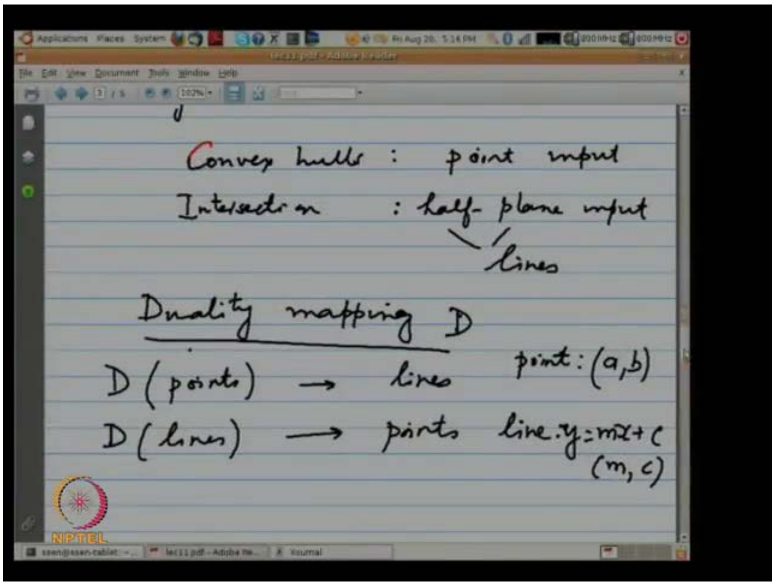
Module No. # 05
Dual Transformation and Applications
Lecture No. # 02
Intersection of Half Planes and Duality (Contd.)

Welcome to lecture 12 of the course; we had spent most of the previous lecture discussing what the desirable properties of this duality transform are. Just to recap, we are trying to solve this problem of intersection of half planes and somehow we wanted to use our previous - prior - knowledge of constructing convex hulls in 2 dimensions.

(Refer Slide Time: 00:31)

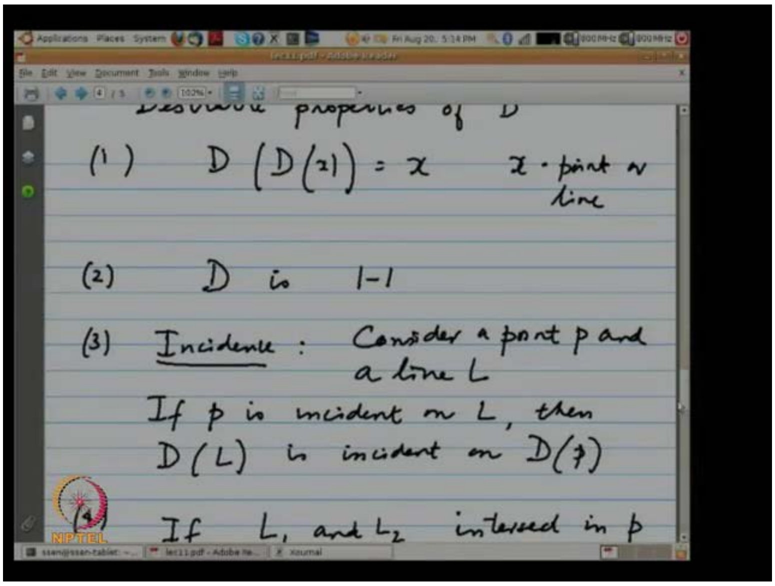


(Refer Slide Time: 01:00)

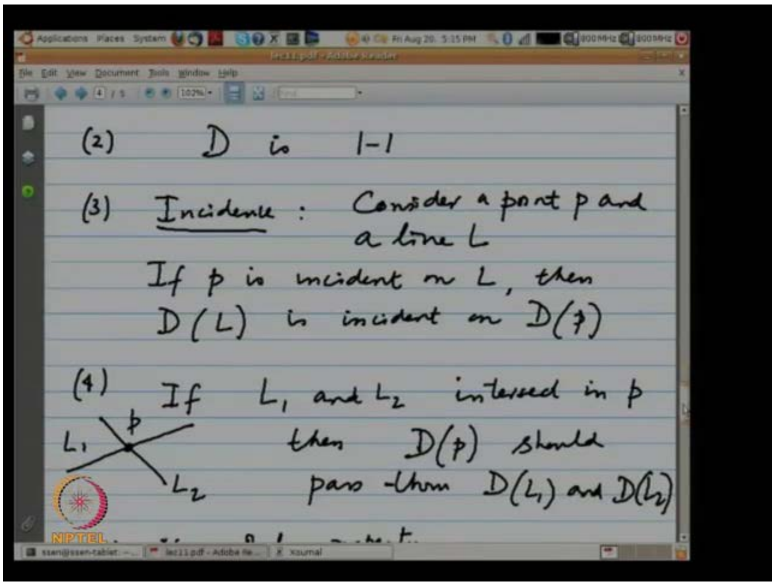


So, we just hypothesized that there is this duality mapping D which maps points to lines and maps lines to points such that these properties are satisfied - that is, it is self inversed, it is one is to one and this is this thing called incidence property that if a point P is incident on line l then the dual of line l which is a point which is incident on the dual of the point P - which is a line; from that the one of the consequences is that it l 1 and l 2 passed through a point P in the primal space then the dual of these lines - which are points - will lie on the line - that is the dual of the that intersection point; that follows from the previous property.

(Refer Slide Time: 01:23)

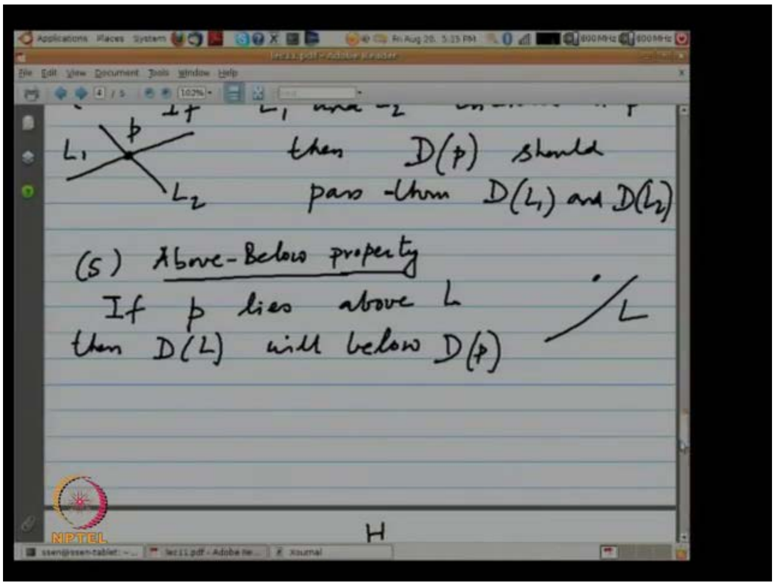


(Refer Slide Time: 01:34)

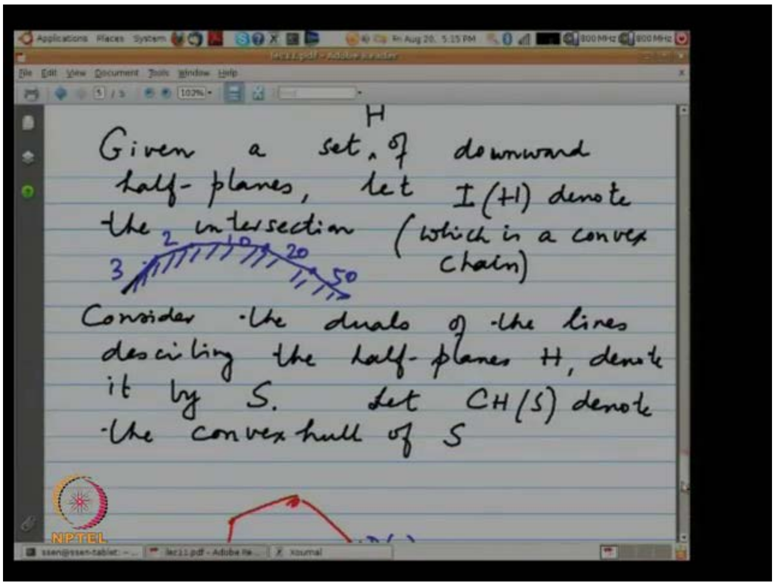


And then there is another important…this orientation property, which is called the Above-Below property - that if a point p is above l then the dual of the line will be below the dual of the point.

(Refer Slide Time: 02:00)

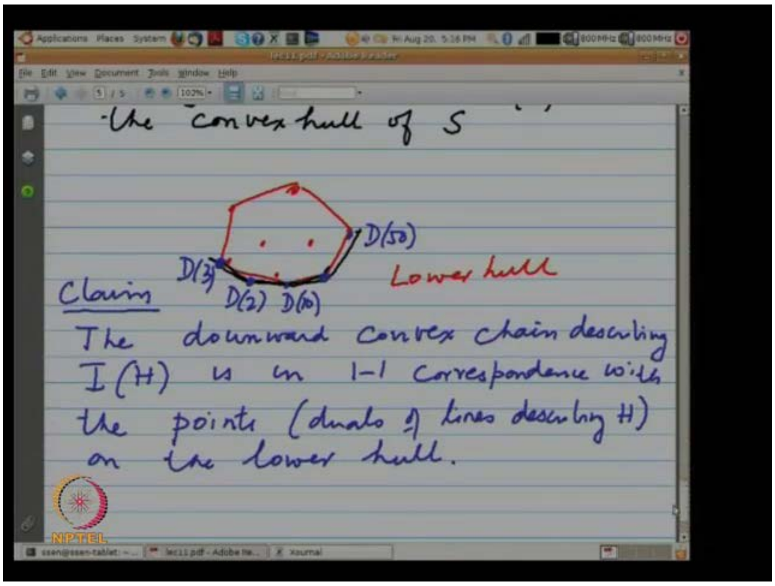


(Refer Slide Time: 02:17)



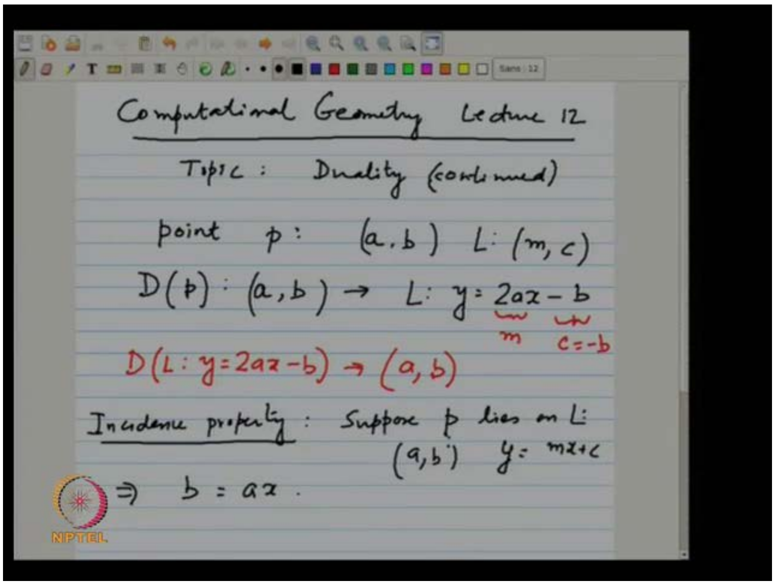
Using these properties, I could show you that there is a close and intermediate connection between the problem of computing the intersection of this downward half-planes - not the entire set; we partition the set of half planes in to a downward and upward; the intersection of downward half planes a has a one to one correspondence with the lower hull of the dual of those lines described in the half planes.

(Refer Slide Time: 02:52)



This is where we stopped last time; everything was fine except that I did not give you any concrete examples of such a duality function - let me do that today; the kind of duality function that you know…there are a number of such duality functions that satisfy the properties that we mentioned; the one that I will be picking up has some special significance which we will see again in the course of this lecture; what we will do is the following: the point again is a pair of coordinates; a point P is some kind of coordinates you know - let us say, x prime y prime; the line is specified in the slope intercept form so it is the slope and the y intercept.

(Refer Slide Time: 03:37)

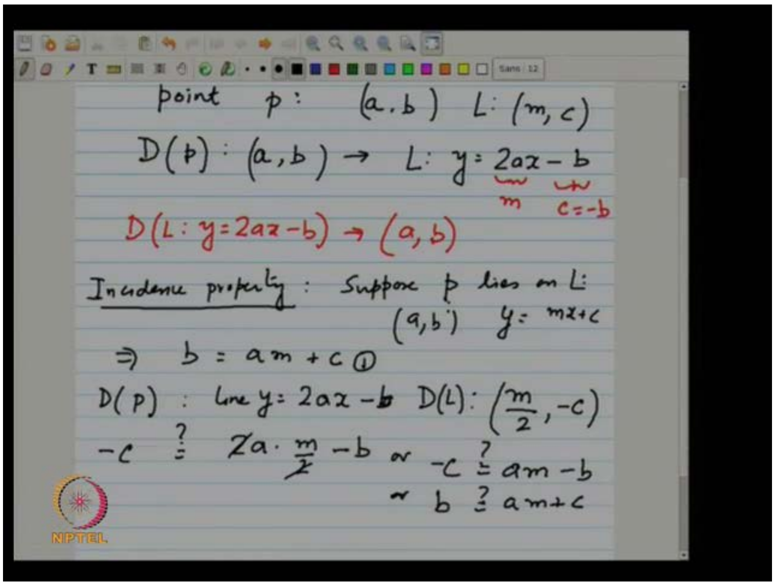


Now, these of P I am defining as - x prime y prime is mapped to…actually let me change the notation so that there is no confusion later on; I will let me use a b instead of x prime y prime; because, normally you will use the parametric equations in x and y directions; let me change that to a and b; a and b will be mapped to this line l given by y equal to 2 a x minus b; in other words, this is my m and minus b is equal to c and vice versa - y these of l given by y equal to 2 a x minus P will be mapped to minus b and vise versa - that y D's of l given by y equal to 2 a x minus b will be mapped to the point a b. This by definition becomes a self-inverse.

Let us try to verify at least.. Self-inverse properties are verified the way I have defined; let us try to verify the incidence property. Incidence property… suppose P lies on l; l is my parametric equation m x plus C and this is given by a b; it satisfies that, which means that this A B must satisfy this equation; so, implies a x… We just (( )) what am I saying (( )).

Fine, that was a confusion I missed that y, which means that…Implies B is equal to a x; sorry, what am I saying - a m plus c, right? Now, what is the dual of the point? That is the line given by y equal to 2 a x minus c… sorry minus b; D of l is given by m by 2 comma minus C - is that right? This is now a point and this is now a line.; let us again substitute; does it satisfy - the question is, is y...Sorry, y is minus C, right? Is minus C equal to 2…this is my equation 1 may be (( )); so, 2 a x is m over 2.

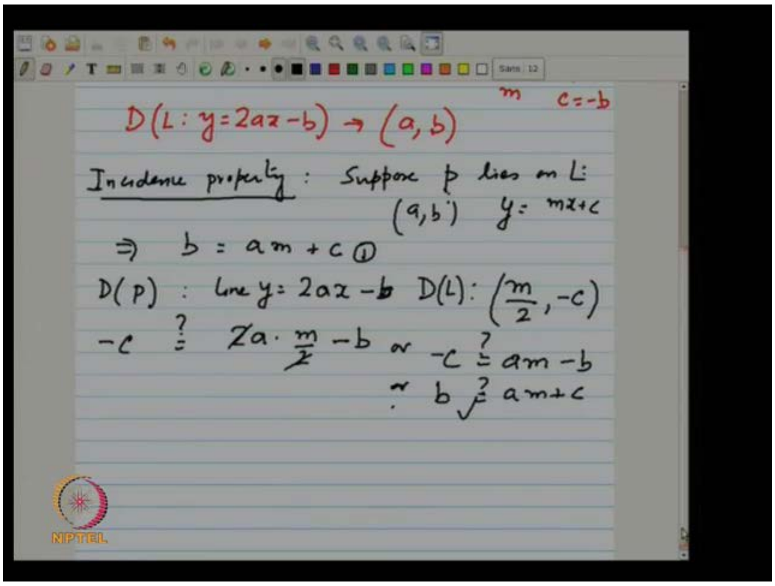
(Refer Slide Time: 10:22)



That is all right…m x the line is minus b…is it equal to this? Or this minus C equal to…well, I am just…I will just translate…anyways, a m minus P is B equal to a m plus C which exactly the same thing as equation 1, so, this is certainly true; the incidence property is satisfied by this mapping; I will leave…it is an exercise for you to verify the other properties - the above-below condition. Other properties basically means the above-below - the property 3, which is the 2 lines meeting at a point will become 2 points lying on the line.

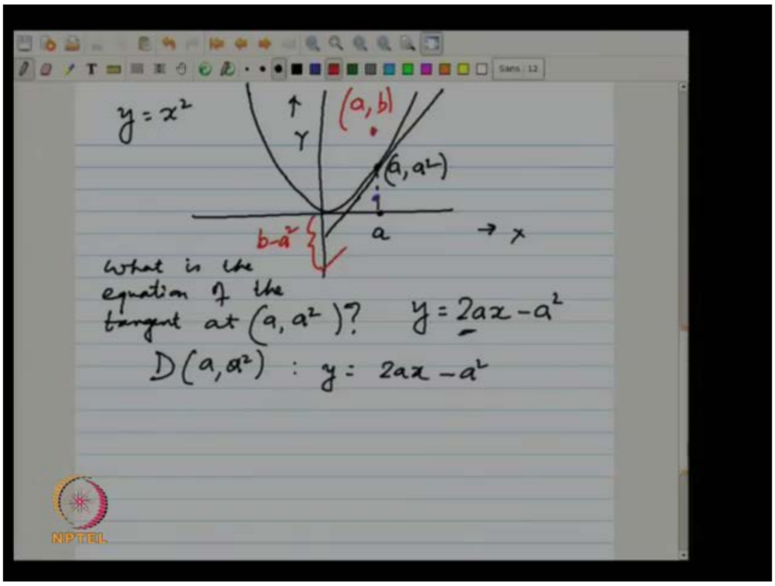
That follows from this property anyway, so that need not be proved separately; prove the above-below property strictly from this mapping.

(Refer Slide Time: 10:58)



At this point it just looks like some symbols - you know some symbol pushing and somehow magically it seems to work out and assuming that even the above-below property holds, but what is a more initiative thing? For the more initiative thing let me draw your attention to where this transform is coming from.

(Refer Slide Time: 11:35)



Consider…let me go to a fresh page; consider this parabola y equal to x square; consider some point a and the x axis - this is x and this is y; a, of course…this point corresponds to a square; now, this particular parabola.…What is the equation of the tangent at a a square? Can you work on it quickly? Remember enough of your analytical geometry.

No, just tell me the answer; I do not want to know how you calculated it; what you are saying y equal to 2 a x minus a square; great, you remember it pretty well; does it now ring any bell? This transformation that…The duality transform that I defined has this….This point a b is mapped to 2 a x minus b; so, at least…Why are we using this constant 2 here? I could have simply written something else; why…This 2 is kind of explained by this 2.

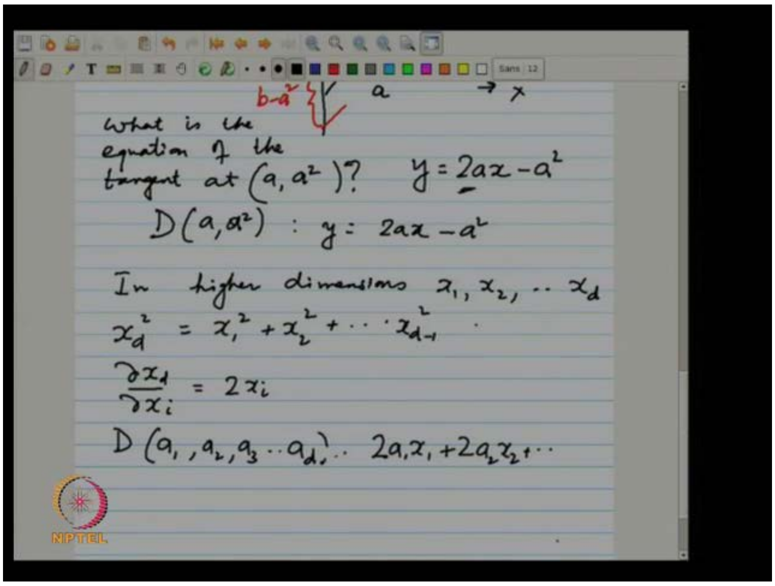
What are we saying? What we are saying is that if the point is on the parabola….That dual transform of that is what? The tangent itself; what is the dual of a a square? d of a a square; by your definition it is y equal to 2 a x minus a square; it is not a coincidence. So, the tangent at a a square is the dual transform of the point - if the point is on the parabola. How about points not on the parabola? Point not on the parabola is…The point could be either above or below, right? That could take a to the same x coordinate - one possibilities of point is above.

The other possibility is that the point is below - may not be on the parabola; it is the same slope - that is a way we have defined the dual transform - the slope remains the same; the Y intercept changes and if we calculate we will find that if the point is above the parabola the dual transform is going to be parallel to this; first off, let me draw this tangent - this is the tangent if the point was on the on the parabola.

For a point above the parabola you can show that the duality transform - the dual transform - is below the parabola; not only that, it will be shifted by something like…this a comma b it is not b minus a square; this point will map to something that is b minus a square.

Likewise, a point below the parabola will be mapped to a line above the parabola; again, with this kind of the intercept…you just take the sign difference and that way what it is… this is the geometric interpretation of this dual transform; when you go to higher dimensions what happens? Can you guess what the dual transform should be if we use the same kind of idea for the dual transform? Then, you are talking about a…In higher dimensions you then talk about a paraboloid.

(Refer Slide Time: 16:57)



Let me not use y and x, but let me use…let us say x 1 x 2 all the way up to x d; these are my d dimensions and I have these as the coordinates; then, my paraboloids are defined as… again some constant - sorry (( )) I am not let me not (( )) not not the shifted parabola it just parabola let y equal to x square.

Then, if you again look at the slopes it will take the partial derivatives with respect to all the coordinates; so, del x d by del x i is 2 x I, alright? Then, can you guess what should be the dual transform in the high dimensions? D of the point - now point in…is a D dimension point and we are taking about a 1, a 2, a 3 up to a d; can you guess what this should be? The dual transform of this point?

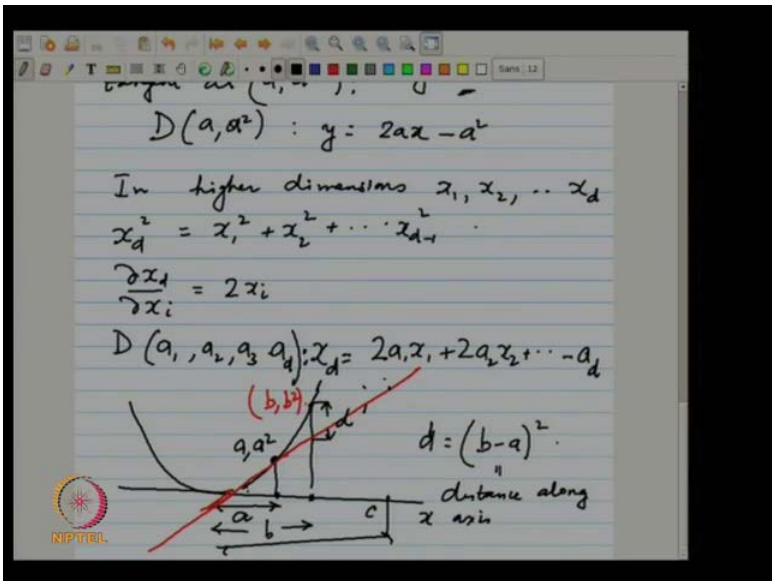
Right, 2 a 1 x 1 plus 2 a 2 x 2 - same thing - plus oh sorry what I am I saying (( )) yeah I have to write that .

Sorry right. This is what it will work out as; again the geometric interpretation remains the same even in the high dimensions; why use this? Well, the same thing - if you want to compute the…We have not studied about constructing convex hulls in three dimensional or high dimensions, but what this says is that if you know how to construct the convex hull in three dimensions you can use the same algorithm to compute the intersection of half species in three dimensions (( )) high dimensions.

Why use the parabola at…. This is a specific transform that satisfies the properties that we mentioned; this is one kind of transformation that satisfies all those properties that we desire from the duality transform; there are also other kinds of transformations that work - there are at least… I know about three or four other kinds of transformations that satisfy those properties and you could use any of those transformations such that you could go from this convex salt to intersections, because all those transformations satisfy the basic properties that we discussed.

This parabola transform… I will just give you another nice property of this parabola transform not just for this thing, but we will be able to make use of it in some even more non intuitive applications; but, just note one thing that here the transformation is not defined with respect to - in 2 dimensions - with respect to vertical lines; because, the above-below does not hold there; there is some singularity or in the dual space of lines one line - the vertical lines - basically have no counter parts; similarly, for any of the other transformations - I am not getting to the details of those - there will be one singularity point and there is a basic fundamental geometric limitation as to why this transformation will have 1 point of singularity; especially if you are familiar with projective spaces this is what…This is also one kind of projective space actually.

(Refer Slide Time: 21:52)



Let me talk about another nice property of this parabola transform; here is my parabola, here is… I am drawing the tangent at a, a square - this is my tangent (( )) let us call this distance which is (( )) b 1 b 2 - this distance I am calling d; what is distance d? I am looking at the tangent at a, a square and I have a point b - this is b, b square and I am looking at the distance of the parabola at b, b square - the vertical distance - to this tangent which I am calling d. Now, d is equal to how much? b minus a whole square b minus a whole square this is just you know follows from this from.

This thing can be interpreted as the following: if this point - I am talking about 2 points now; you know 1 is this - a a square another this is - b b square.; if we look at the projection of the points on the x axis, namely this a and b, the distance between this is b minus a; this is the distance projected or otherwise this is the projected distance on x axis - whatever - distance of…As if I have just projected those points in the parabola to the x axis - that is the distance.

That distance square is equal to d; what it says is that further the point…if I take another point c….Basically and I look at well I mean… Again, the vertical distance of c, c square to this line it is sort of directly proportional to the projected distance on the x axis.

We will just remember it as the following: if the distance on the x axis between 2 points is more, the vertical distance that you obtained from the parabola to this point also grows - it grows as some square, but essentially it is proportional; the larger this b is the larger the distance; in future when we talk about things like finding closest neighbors and discuss about voronoi diagrams you will see that there is a very close link using this transformation between what is called the three-dimensional convex cells and voronoi diagrams.

Like we could get these for free that you know this - two for one deal - convex hulls and intersection of half planes; it turns out that there is a very similar relation between know a 2-D voronoid diagram and a 3-D convex hull; now, I will just leave it that and I am drawing your attention to this property; only this property will be used to…and this works for any kind of….I have moved from x…The projection is along x and the parabola is a two-dimensional curve.O

If take the set of points from x and lift it to this parabola - which is one dimension higher - essentially what I am saying is that in general if I take these points on d minus 1 dimension and project it to a parabola which is a d-dimensional paraboloid, then I have some kind of relation between the distance in the D minus 1 and the vertical distance from this parabola to the tangent at this point; if you are looking for…. I am giving this point a - this point - and I am trying to find the closest point that has clearly some relation between this in terms of the vertical distance of this point.

Whatever is the distance - distance between a and c or distance between a and c can be captured - at least the relative distances can be captured by this vertical distance from the paraboloid; that leads to…we will be - I will do it more formally later on; this exact one to one correspondence between voronoi diagrams and convex hulls, but with one dimension gap - three-d hulls and two-d voronoi diagrams - that kind of thing; therefore, you know when we study what is called the structural properties of voronoi diagrams that follows from the convex hull in one higher dimension; this really sort of simplifies a lot of things that people study about voronoi diagrams.

In general, what we study will go beyond this; I will talk about what are called arrangements of lines and arrangements; that has basically…we will all pull it together later and show you how they all correspond; for now this is all I will not talk about duality; any questions at this point?

In the higher dimensions does this points transforming the lines and vice versa, but what about (( )) there is no conformation for (( ))

No no. The transformation is defined for all dimensions, right?

No, but is only points going to lines

Points going to lines and lines going to points.

And like if you have a face that that is not transformation of the power you just apply individual.

No no, essentially it is just a point hyper plain transformation, right? So, if you are talking about a d-dimensional space you are mapping it to another d-dimensional space;I am calling the primal space as we call the dual space; so this particular duality mapping is a point hyper plane transformation; there are other kinds of transformation people talk about, but for the kind of problems we are looking into this point hyper plane transformation is the is the one that is most important - very useful to us.

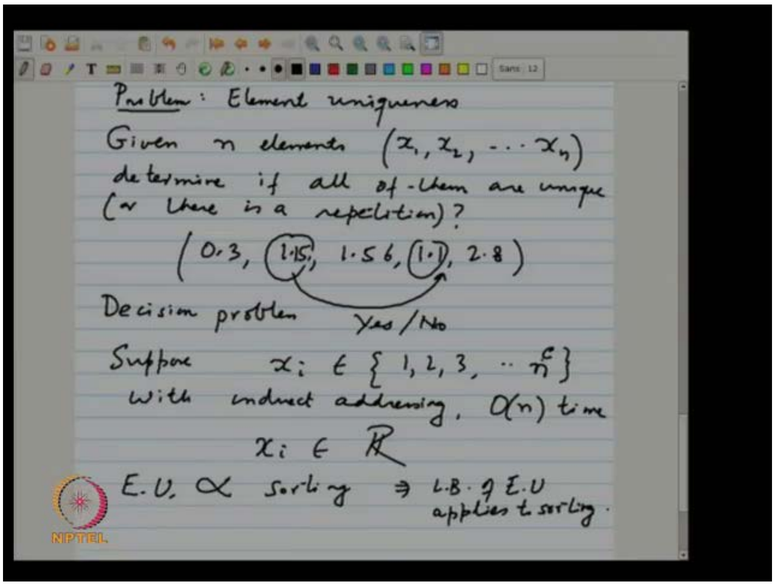
This preserves the dimension in some sense; all though they are 2 different spaces, always remember the points - the space points - the space of lines they are not the same space - it is the dimension of the space that is the same; we are not talking about the same space - one is the space of points another is the space of lines or space of points and space of hyper planes.

The other kinds of transformations we did are where this preservation of dimensions is not necessary; things like when you go from segments which are actually… see it is basically you know how many coordinates do you require? How many parameters do you required to capture the objects? Think about a line segment - how many parameters do you require to capture a line segment?

2 points, right? Which basically means 2 coordinates each so it is actually like a four-dimensional object; there actually transformations between lines and some kinds of curves in the five-dimensions - is called Plucker Coordinates.

There are various such things that people have studied in geometry and some of them we know are useful for doing computational geometry. Since I was already talking about four and five dimensions I will spend the rest of the lecture today and probably the next lecture also, which I have sort of I have been postponing for a while is talk a little bit about how lower bounds are derived in geometric problems.

(Refer Slide Time: 31:47)



Let us use one running example; right now let us worry about this problem - let us call it problem; if I am given a point set like - sorry - a set of elements which have the following values 0.3, 1.1, 1.56, 1.1 and 2.8; these sets of elements clearly have a repetition because these two have the same values.

So, the answer…this is a decision problem; decision problem, which has a yes or no answer; if you consider this particular instance then the answer is no, right? It is not unique.

On the other hand if I change it to 1.15, then the answer is yes - all elements are unique; of course, these elements can be drawn from all kinds of universe; if the elements… suppose xi (s) are from this set, what would you do? What is the easiest way?

So, the easiest way is do it is basically hash; I have an array of size and corresponding to the value of each element I hash and after I hash I find out....Essentially indirect addressing; with indirect addressing it takes (( )) n time, fine; what if I do this? Why? Well, at least I made you think; you can hash, fine, but after hashing…first of all when do you…When you hash there is one problem about hashing the people always forget - how do you initialize?

There is a space in…Hashing is about using some space, right? That space somehow needs to be initialized; if you are hashing in n square space that is a problem; okay, you just use n locations; so, what when you use n locations what are going to do? Well, you are getting into somewhat non trivial things - there should be should or should not be collisions, how many collisions will be there; well, if you have done a course with me in algorithms you may have…Someone would have got a doubt - use universal hashing does it (( )) I am not even going into get to those kind of scenarios; but I claim that there is an even simpler way think about it; no, no…So that you can forget about parabolas…n to the power 5.

Someone should (( )) answer by now; do not always think about linear time solutions and I am not saying that you have to give me linear time solutions, but what would be a natural way of solving this problem? Yeah, right, I wanted here that; if you sort them, then, of course, any repetition will be in consecutive location; after sorting it can be checked in linear time.

Now, if I want, can I sort these elements quickly? No, come on; right, you know the range and it is a polynomial range, so, what do you do? Yeah, well you call it various things and I call it radix sort; radix sort means for any range, which is polynomial you can do it in order in time; are you familiar with this? You just run your buckets from 1 through n and you repeat c times and it has to be a stable sort to be able to apply radix sort (( )).

People always forget about radix sort I do not know why; you will remember merge sort, in solution salt, all those useless things really - not merge sort, but insertion sort, but some of people never think about radix sort; I emphasize so much about radix sort when I teach you algorithms and still people (( )) people who have done the course. So, you cannot do radix sort on this and there is a sorting is stable we can take it offline in the class.

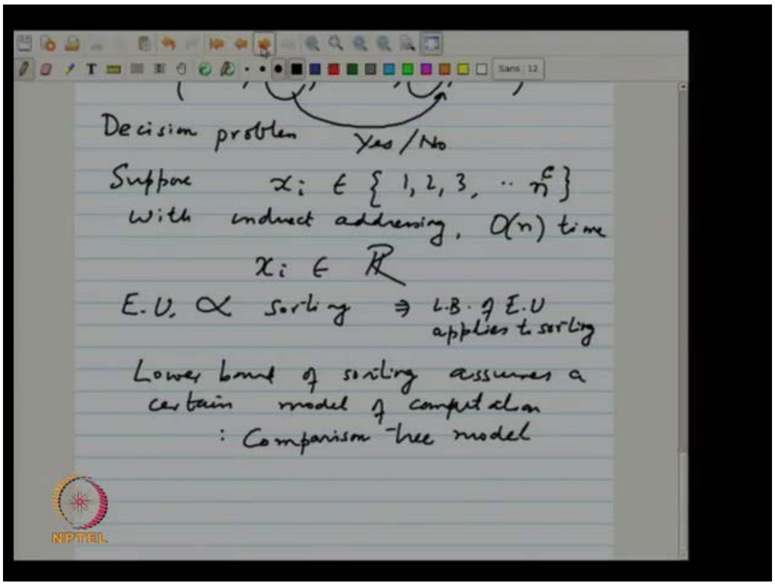
No, no, no - radix sort is not a sorting algorithm per se; radix sort is a high level sorting algorithm that uses a basic sorting algorithm, so that sorting algorithm must be (( )) .

Order and time is x i's polynomial, but what if suddenly I just blast it and no finite - this thing absolutely infinite set no.

Yeah, now you can sort it; sorting is reasonable, sure, and it will give you whatever the sorting takes; essentially, what you are saying is that yes I can certainly solve elements uniqueness in the time that is required to sort; question is, can we do better? So, element uniqueness - let us call it e u - is reducible to sorting.

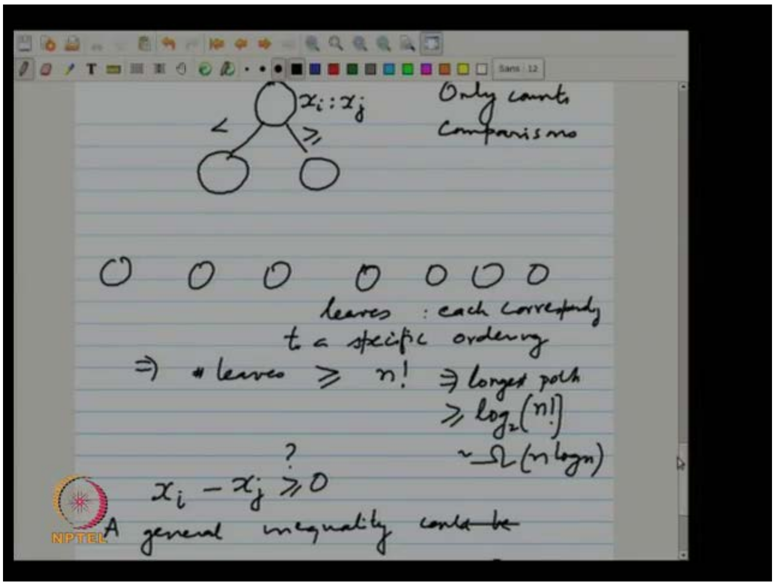
That is what we are saying, which means that the lower bound of element uniqueness should also apply to sorting; but, then we are actually interested in the lower bound of element unique - it is not about sorting; we cannot apply the sorting lower bound to element uniqueness, but you can do it other way round; now, all the lower bounds - not all lower bounds - the lower for the famous lower bounds or the most commonly known lower bounds for sorting assume a certain model of computation.

(Refer Slide Time: 41:05)



And, what is that? Comparison tree. What is the comparison tree model? Well, I mean since it is not a decision problem it is more than that, so I am using the word comparison tree rather than decision tree.

(Refer Slide Time: 41:40)



You compare some elements x i x j and branch according to less than or some such thing; depending on this thing you keep doing it untill you get your leaves right; leaves - each corresponding to a specific ordering; therefore, number of leaves should be greater than or equal to n factorial; a leaf cannot have to possible orderings because…And for my argument you know I am going to use greater than equal to - it means greater than ‘or’ equal to also means equal to right greater than or equal to. So, number of leaves is greater than these and this implies that the longest path in the tree is at least log of this.

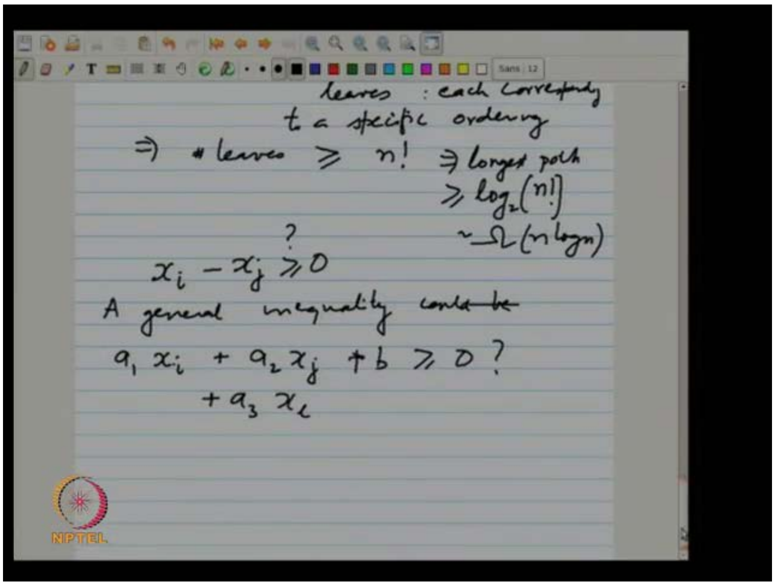
Longest path, which is about (( )). This is your basic argument for the lower bound for sorting and….The comparison…and we do not actually count anything else….The comparison tree model only counts comparisons - only counts comparisons; this tree is only to count comparisons; if you are doing some additions, some other things like assignments, whatever it is - other kinds of operations are not being accounted for here, but let say that you are basically using comparisons to sort you are stuck with this bound; this is something that I have been alluding to a couple of times that why should one only limit oneself to comparisons; what is comparison? Comaprision you can think about like I take two numbers x i minus x j - that is basically my comparison.

This is a special case of an inequality - a linear inequality - where both coefficients are 1; say a most general inequality could be - this could be another test, it is more general than comparison, but it is still linear; why even have two x i x j you know x i x j x l - may be I can bring in more dimensions.

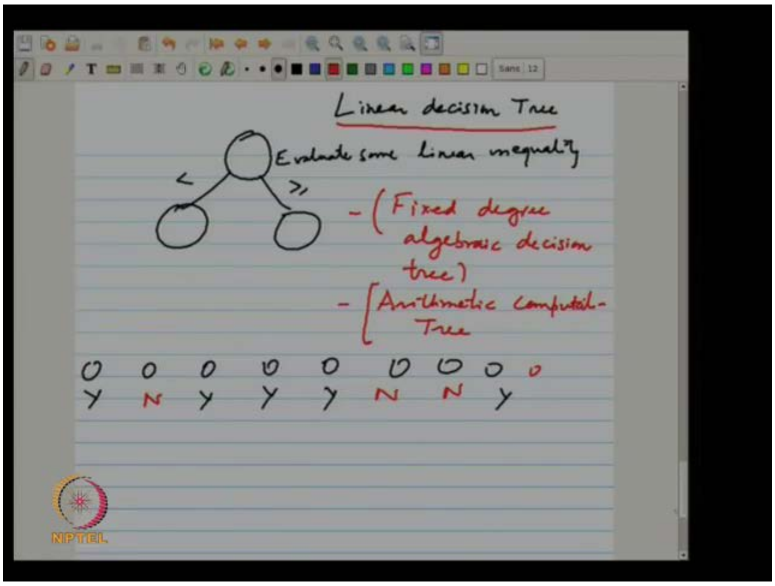
Now, it is looking like - not like just comparing to elements - but in some dimensions we are evaluating this inequality; in other words, are we above the hyper plane or below the a hyper plane? That is basically the kind of question; the moment I do this kind of…I say this is the operation that I am allowed to do in one step - the argument that we used previously is not going to - is going to - fall apart.

But, because if you have done the argument rigorously it was only about these comparisons; somehow, we need to redress and you know and see if actually those that lower bounds hold in this kind of a model; right now we are not talking about sorting, we are taking about element uniqueness; the model for elementary uniqueness - sorry.

(Refer Slide Time: 46:34)



(Refer Slide Time: 46:59)



It is what I call a linear decision tree; a decision because we are discussing a decision problem namely the element uniqueness is yes or no; linear because I am allowed to use only linear inequalities; so, at any node I evaluate some linear inequality and again depending on less than or greater than or equal to i branch; in the end, in the leaf nodes the answer is either a yes or a no, right?

May be yes yes yes yes yes no no no etcetera; how many leaf nodes should there be? This is not sorting; how many leaf nodes should there be? Not clear, right? Certain number of leaf nodes - hopefully some kind….Any algorithm that uses linear decision - sorry - any algorithm that is using linear inequalities to make the - to decide on the next step can be captured by this model, which I am calling the linear decision tree model.

There is this problem which is element uniqueness and there is this problem; then, there is this model which is a linear decision tree model and we have to somehow link up this problem and this tree; somehow the structure of this tree - like in the case of sorting the length of the problem was sorting with some kind of structural properties of the tree that had at least n factorial leaves; here we cannot even ague about that because it is some kind of a decision problem; we cannot say there are n factorial permutations or something.

We need to be able to draw somw linkages between the problems and the tree - that is how lower bounds are going to work - I mean work out; let me also just make a small remark here - I am discussing currently linear decision tree, but similar arguments of course, with you know more….We need some more tools we be able to handle that; you can model what is called an algebraic decision tree where the basic inequality may not be linear, but it may be higher degree - maybe 2 degree 3 degree whatever.

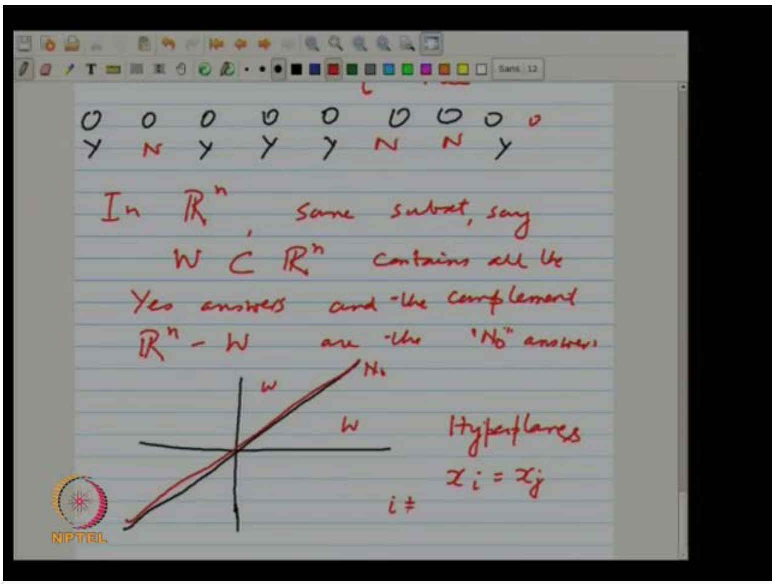
Those are called fixed degree - I am just mentioning - fixed degree algebraic decision tree; that is one variation other variation is in more general; why even fix the degree? I can actually…when I do computation I can raise number to any power that I want - why should we be constrained? I can raise it to any power by repeated squaring, which means that in a very small amount of time I can really sort of (( )) of the degree; I can go from x to x square to x square square x square x square. So, in i steps I can compute x to the power 2 to the power i.

So, why should we limit ourselves to only fixed degree? I can actually compute a x to the power n very quickly in polynomial time; then you talk about what is called the most general is called arithmetic computation tree; I am not going to discuss these variations, but it is somewhat closely linked to the discussion that we had.

Today let me just talk about the problem space; the problem space that we have - the input is this tuple - this n tuple, x 1 to x n; now, for a moment just think about the input as a single point in n dimensions; we have an n tuple and can consider this n tuple as a point - as one point - in n dimension space.

What is it that we gain by that? Not much, but actually it makes…although it looks complicated it makes thinking somewhat cleaner; in that high dimensional - n dimensional - space certain points are yes points, where all elements are unique and all coordinates are unique; some of these points are not all unique and therefore they are the no answers; in my space of… What is the space I am talking about…I'm actually taking you to r to the power n.

(Refer Slide Time: 52:05)



In r to the power n some subset - say, w - contains all the yes answers and the complement are the no answers; whatever my algorithm is and if I am modeling it as a decision tree in the end any point that I input - any point that is in (( )) - eventually should end in a in a leaf node; if is a yes instance it should end up in a leaf node that says yes; if it is a no instance where not all elements are unique it should end up in a leaf node corresponding to no this thing the label of the leaf node; just to give you a very simple example in just 2 dimensions, how does this space look? What is the space…what is the set of points corresponding to yes and what is the set of points corresponding to no?

Yeah, right; that is it, right? This is my no and this is basically w; in the plane - if you are talking about a plane you can visualize this - as you go to higher dimensions you have to basically look at the intersection of these half planes.

There are some half planes or you call them half hyper planes; you have to look at the hyper planes x i equal to x j, i not equal to j; in n dimensions these are the hyper (( ))

these are the hyper planes that will contain those points that will correspond to the no answer - i not equal to j - and you have to look at what is called essentially the arrangement of these hyper planes in this n dimensional space.

Union of all the points that lie on the hyper plane correspond to the no answer and the compliment of this corresponds to the yes answer. Any point that is not on any of these hyper plane is an n dimensional point; if the n dimensional point does not lie in the union of this hyper planes it means that all coordinates are unique; let us stop today and we will continue with that next time.