Computational Geometry

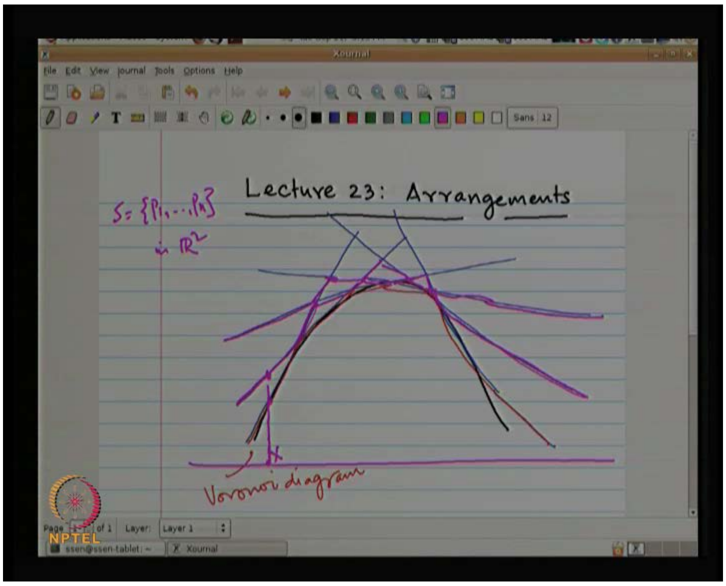
Prof. Pankaj Aggarwal
Department of Computer Science and Engineering
Indian Institute of Technology, Delhi

Module No. # 10
Arrangements and Levels
Lecture No. # 01
Arrangements

So, I will continue where I had stopped a week ago, but I will switch to another topic soon, because it is related to what I was talking about in terms of Voronoi diagram. So, just to remind you, I introduced the problem of computing Voronoi diagrams to computing a minimization diagram of a set of aqua planes. Am I right?

So, the problem looked like you had a parabola and so these are the planes. If I look at the low envelop, so if I draw the low envelop and project the low envelop that was a Voronoi diagram.

(Refer Slide Time: 01:08)



If you remember the definition of Voronoi diagram or the low envelop, this was the set of points. It is a point wise minimum. Am I right? Now, what I do? Suppose, I play the following game that instead of remaining the lowest, I remain above one line and let us try to trace the path. So, let us do the following. So, look at this purple line, purple polygon line that I have drawn. What is this corresponds to? So, if I project this purple line on a plane, so other way we are thinking about if I take a point here, let us call it x and I look at this lines. So, these lines, each of these line if you corresponds to one of the input points, so you had the set of input points.

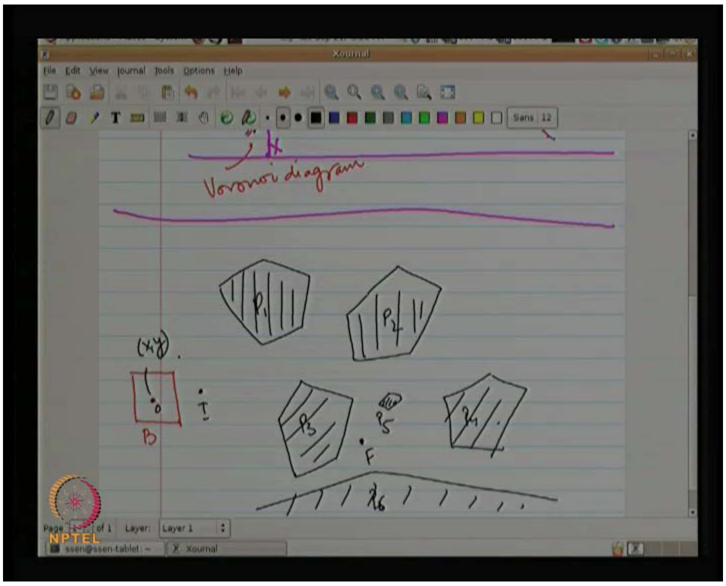
Well, I have drawn in one dimension and then, from this point, I had come with the cones. I think this transformation, I get the lines. So, planes into 3d then. So, this place, each of these places corresponds to a point minimization diagram. The Voronoi diagram corresponds to the low envelop. If I look at the one line above does not even have sense. What this corresponds to? Pardon. So, suppose what does this correspond to like suppose if I look at this point, the lower envelop, what is this? This is Voronoi diagram.

So, what is the Voronoi diagram corresponds to nearest neighbor? Now, if I go, so this is the second nearest neighbor. So, if you look at this, what this gives you? If you project this purple line, what it gives you is what the second nearest neighbor of every point is. If you want to, you can continue doing this way. So, if you, similarly you should do the third line. This corresponds to a third nearest neighbor. So, you can talk about not in the nearest neighbor, but one can talk about the nearest neighbor, second nearest neighbor and so on.

Now, I can take to the extreme. If you look at the upper envelop and because I defined the low envelop as a point wise minimum and upper envelop is a point wise maximum. Then, what does upper angular correspond to? Yes, it is the farthest neighbor. So, just keep this in mind. I will come back to it. So, this notion of beyond just the low envelop. So, this is just to keep in mind because I am going to put all these things together in a unified frame work which I will call arrangements and that is what I am leading to go ahead.

Actually about polygonal region, it is still polygonal region by truly much more complex. Now, I will come to it later. Let me completely talk about very different application.

(Refer Slide Time: 06:00)



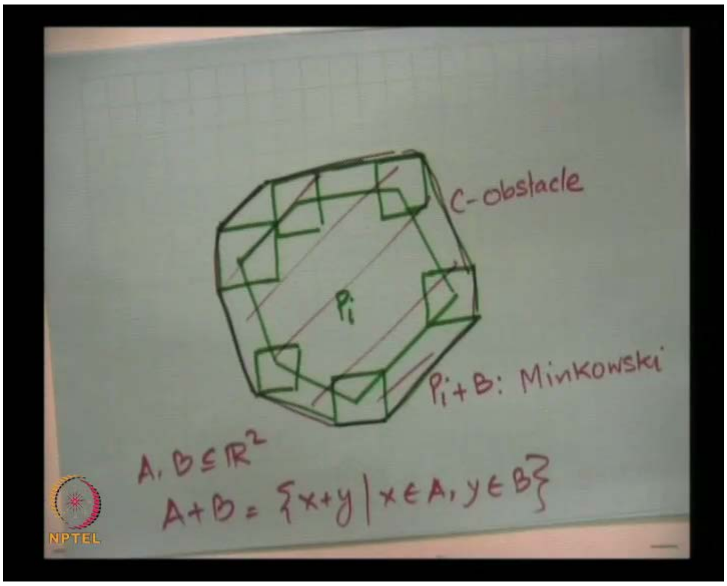
This comes from a robotics in a motion planning. Suppose, you have a, let us say that is why these are some obstacles, some polygonal environment and a robot is a square let us live in then. So, the goal is following you. I will give you some initial position of this square which you think about, which some is modeling as a robot and I want to take this robot from one place to another place. You can only translate it, you cannot rotate it.

So, think about this is a puzzle now. Of course, in general, the robots of the much more complex shape and it is not just you translate and rotate. Sometimes, they have arms and you can move. So, like human body if I think about has many joints, but this is just a toy, example just to illustrate the idea. So, think about it that if you want to figure out whether I can go, let us say from this position to well let us say this whole thing is this. This is something here also.

Now, you can go from some place to another place. So, let us say this is an initial place that you want to take it and this is let us say, the final place placement. Now, cells at robot, this square, in this case can only translate. I can represent its position. Just choose a center of this square and tell you what its coordinates are. So, I fix the center. Let us call this a center point as O. I just specify its coordinates x and y. Am I right?

That is must have specified the coordinates of the x y coordinates of the center. The position of the robot is fixed, because there is, no for you there is no freedom of rotating it. So, does anyone having gas? How will you try to figure out if whether there is a collision field path? Of course, what I should say? When I say, it can go from one position to another position that when it you take it and translate it during the translation, it should not collide with any of the obstacles. So, in this case, you should not collide with any of the polygons I have drawn.

(Refer Slide Time: 11:40)



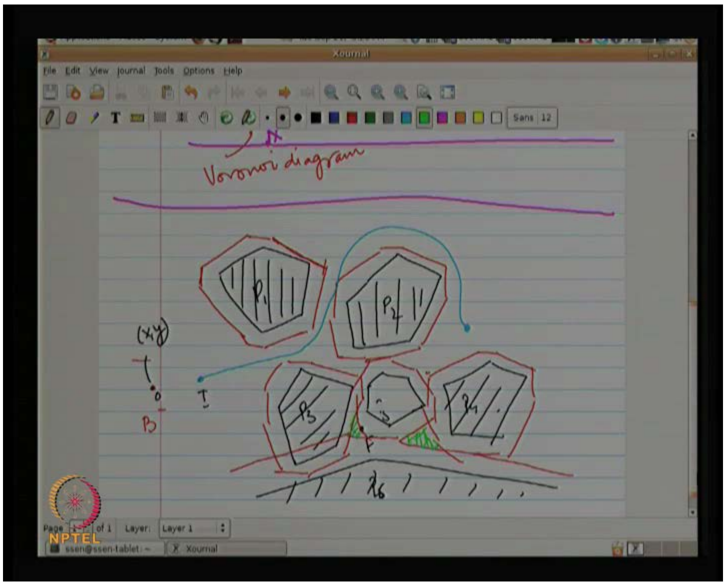
I am just asking you the decision co-efficient whether it is feasible. Now, of course, once you can talk beyond the feasibility question, you can talk about after addition problems. What is the shortest path or you want to stay away from as far possible from the obstacles? So, it can have if you think of this robot as a car, then you cannot turn very sharp and so on, but let us keep all those things aside from. Now, listen just very simple problem. So, how will you do that?

So, the Voronoi diagram, now the problem with the Voronoi diagram if it is a disc, then I could have done a Voronoi diagram. Am I right, because the distance here is little tricky but you raise a set of an interesting approach and if I have time, I will talk about it. There are Voronoi diagram based on approaches and I will come to it. If I have chance, I will talk about it, but you are right. There we see the question is when you think about something, one can do behind the right direction that what you need to free out is the path that is still away from the obstacle.

Anyone else wants to make a guess? So, I want to introduce a very general technique that lies at the heart of all the motion planning problems in robotics. This notion is what is called the configuration space of c obstacle and here is a trick that is used. What I am going to do is I am going to expand the obstacles and shrink the robot to a point. So, I will not formally define it today in general, but here is what you do. You think about the following that you think of taking this is square that you have. So, I think to use a form. If I do it here so one thinks about the following. I take the square and I put it center line it to the boundary and I just trace it along the boundary.

So, you think about I just take this square and I drag it along the boundary. When I drag it along the boundary, what will happen? It will sweep some region and you will get some larger polygon. So, let us do that. So, you got this bigger polygon now. So, let us, this was polygon p i and this is called p i plus b, bigger of it. So, this is the notion. This is called c obstacle.

(Refer Slide Time: 13:25)



Now, horizontally the polygons were not intersecting, but let me do it for every polygon, but this is boring. So, let me do something. Now, what you notice is that all the additional polygons when intersecting, once you expand it, then they start intersecting ok. Now, if I expanded this one, then what I can think about we can shrink this robot to a point. So, you can think about this and I question you now in this case, everything is still connected. Well, not quite actually. I forgot to do the last one. So, for example, if I want to go from here to here, then there is a path right. The point can go here to here, but now if I want to go from here to here, there is no where you can go because there is no path.

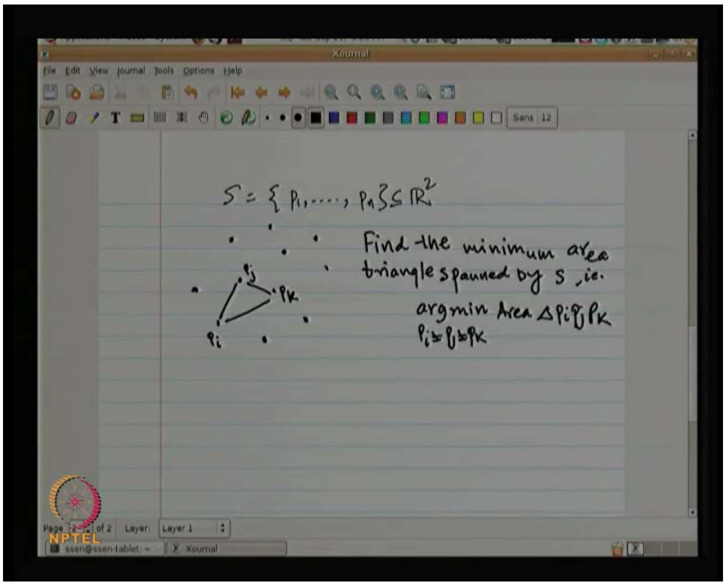
So, the problem base can or reduce is to after you expanded the obstacle, shrink the polygon to points two point as you ask the question that the initial position and the final position where that they lie in the same connected component of the space. That is not covered by the expanded obstacle. So, for example, there is one outer component and then, these are few timid disconnected components. Is that so? You cannot go from outside to here, but within if you want to go from here to here, you can go. You can do that.

So, both of this, what is problems basically happening here also, what we did was, you had some polygon, some lines and you expand it that you got genuine polygons and you took the union. Look at the compliment of the union. So, this is called a configuration space and minkowski. So, this p plus b, I wrote this is the mathematical term for, this is Minkowski. In general, if you have any two points sets a b r square or general in any dimension, think about a set of vectors. In both of this problem of Voronoi diagram or the motion planning application I talked about, there is a common theme.

The theme is that you have a set of lines or set of polygons. Then, you draw them in the plane and partition that plane into some regions. In this case, some of the regions are interesting regions like many and the low envelop point k1 envelop or in this case, some of the regions are interesting some regions are forbidden. So, this is let to lot of work on understanding that decomposition in due spy set of geometric objects.

I will start with the most simple geometric shapes set of lines, but before I do that, I want to also give you one more problem to set you. Think about that. We also come back to it to show how all these techniques help you to solve. Let me ask the following question. So, suppose you have given a set of points. Now, if you take any triple of them, if I take any pip j p j, let us say this may be a triangle. Am I right?

(Refer Slide Time: 18:33)



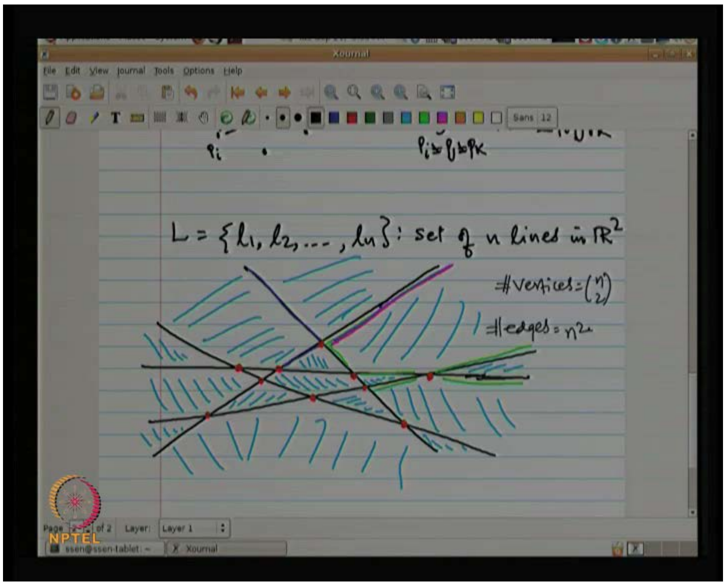
So, there are n choose three triangles because the n points are n choose three triangles. I want to find a triangle which has the minimum area. Find the minimum area. So, find the triangle that has the minimum area. What is the simplest way of doing it? How much time will this imperial got into it and cubic time over that because just try take all triples, compute the area of a triangle and choose. Anyhow, if I ask you, can you do it in quadratic time?

That is a good observation. How will you do that? How will you find such triangles? Pardon, but that will give you only some triangles, not all the possible candidates. No. If you take a triangulation, then by definition there will not be any point, but how will you guarantee that the triangulation you constructed contains a minimum area of triangle. That is why but not necessarily the closest because you can take for example, you can have an example. So, that is up makes it hard this problem.

(( )) it may be the convex hull and the triangulation and then (( )). Yeah, but how do you guarantee that has a minimum area of triangle (( )). So, the difficulty comes is because you may have two points very far from each other, but now suppose, if the third point also lies on the precise, the line its area is zero. Am I right? So, it is not just taking the points close by that is what makes hard. That is why Delaunay triangulation is not good enough because drawing, actually Delaunay triangulation make sure that such a triangle will not be in Delaunay triangulation.

Set out avoids having such triangles doing it. So, one observation that is mentioned that no point should lie inside that solve as a good, but they have still many triangles one has to look at. So, somehow one has to prove such will doing it. So, I will show you again how this will be helpful and exploiting the observation that we mentioned. So, now let me sort of with these examples. Now, let me sort of start formally defining what an arrangement is and how we construct them. Then if I try, I will say you how you solve these problems using arrangements. I will continue next time.

(Refer Slide Time: 22:48)



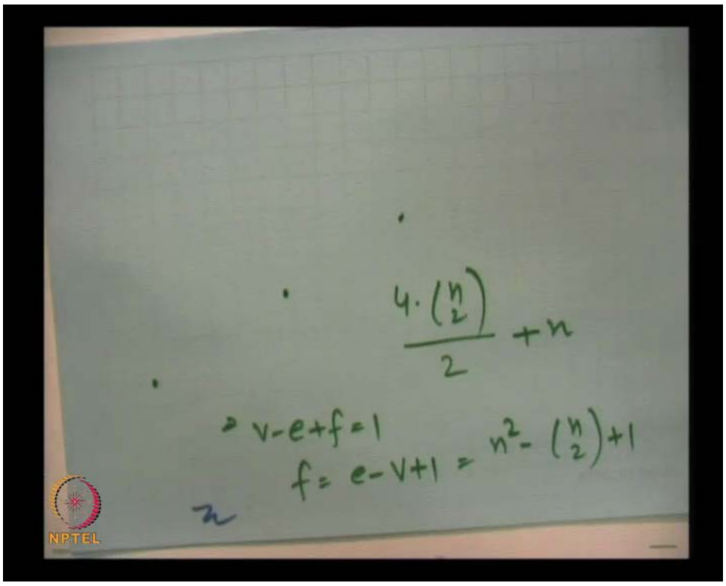
Any questions? So, if I draw n lines in a plane, then you get intersection points with these lines. When you draw them, they form a plane of sub division whose vertices are the intersection points. So, these are the vertices of the planer. Planer sub division edges are the portions, the lines between two consecutive vertices, so simple. This is an edge; this is an edge and so on. I will not draw all the edges and then that portion of the plane that does not lie inside any line connected component that is called a phase.

So, if I look at this is a phase. Now these are the. So, you have vertices edges and phases. It is a plane of sub graph plane of sub division some of the edges and phases unbounded they are some raise. So, for example, this is inbounded edge and some will be bottom and top, these are unbounded phases. How many vertices are there? This is simple. I will just do. How many edges are there? Pardon.

Why n choose to minus 1? So, let us all the degreeth of 4 almost there, but there is little. So, because when you one thing, notice that not every edge because the problem is there are some unbounded edges. Am I right? Unbounded edges, they are not incident to two vertices. So, here is the simpler way of counting for each edge. Look at how many portions each edge is split into each line? Sorry, how many portions each line is split into? Why n minus 1? Why n plus 1? So, look at the fixer line. So, you have n lines, so fixer line.

How many other lines are there n minus 1? So, n minus points (( )) point, so exactly n place is this plate. Am I right? There are n lines. Each of them is split into n portion. It is n square. Now, you can reach around the same thing. The way people are saying that it is a 4 times n choose 2, now you have to divide by 2, because some of the edges have been counted twice, but you have done undercounting because some of the edges are unbounded. You have to add n again.

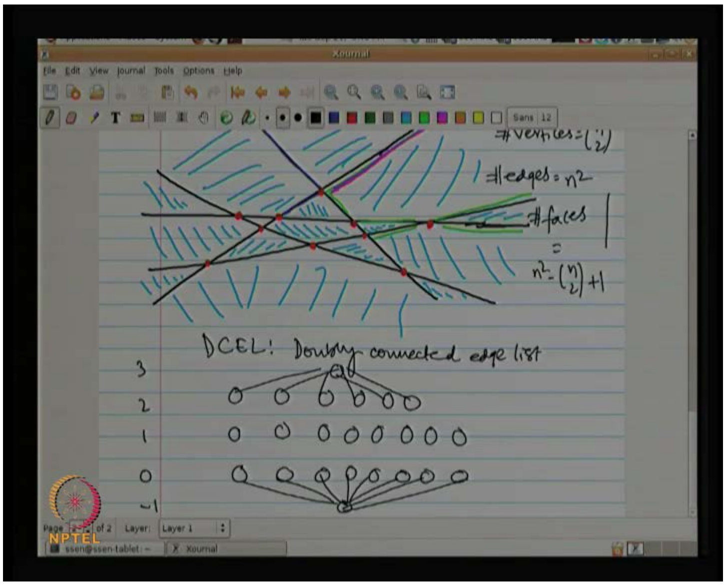
(Refer Slide Time: 27:06)



So, that will bring you back to n square. Now, you can use Euler's formula to count the number of faces. So, number of faces will be since it is unbounded stuff. So, it is a formula is v minus e plus f is equal to 1. So, f will be e minus v plus 1. So, will be n square minus n choose 2 plus 1.

Now, one question you ask is, first question is how will you represent this arrangement? Have you heard the data structure called D C E L, doubly connected edge list? So, that is one way of doing it. So, advantage of doubly edge connected edge list is that you can traverse any face or you can move there on the any vertex. So, along the vertex has is a less you want.

(Refer Slide Time: 28:30)



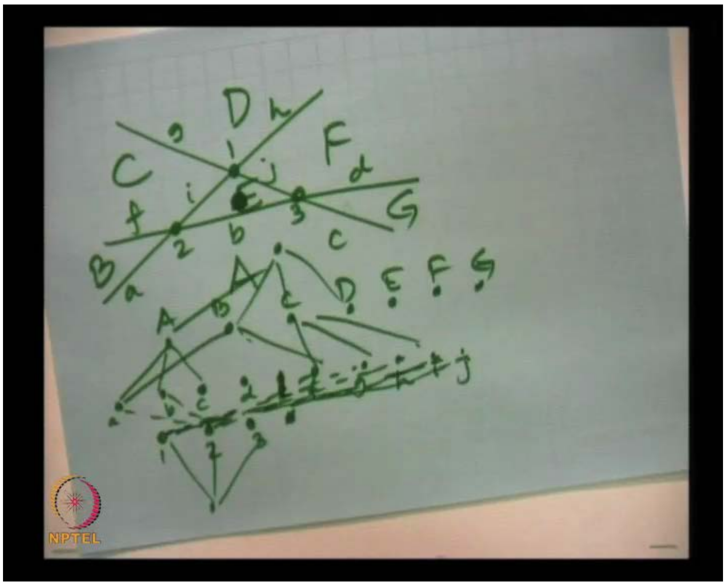
So, you can traverse arrangement quite easily. Now, another presentation that is used quite often in such cases, and some other presentations are called lattices structure. So, let me just show you because I think, it is useful not only here, but for other applications also. So, what you do is, you define. Lattices is a graph, you did even construct a graph.

Where? So, before I do that let me see the following. The word face is going to be abused on one hand. Am I here calling two dimension faces, but also in high dimensions, a face is called what is called a k dimensional face. A vertex is zero dimensional face, edges are one dimensional face and what you called face, here are the two-dimensional faces.

So, first you draw a graph. So, let us see where you have. Let us say, call it zero. This is dimension 12. Well, I am not going to draw the whole damn thing. So, here all you draw the note for every vertex, you draw a note for every edge and then, you draw a note for every face, two dimensional face and 0 1 2 talks about what dimension. Then something you want to have it is common co-vertex in and also the highest vertex you have to move vertices. They are called minus 1 here and here this will be called 3 which will be d plus 1 dimension.

So, this you add to everything. This every vertex is incident to this is what technical reasons and the rule is following. You are going to connect a zero dimensional node to a one dimensional node if vertex is connected to an edge. So, for example, let me just draw a simple example.

(Refer Slide Time: 31:30)



Let us just draw for this star 1 2 3 and a b c connects 1 2 3. So, one is connected to it and I am just saying of f i j and I am going to get into trouble now. So, one is connected to g and h is extra. Let us we forget e. So, one is connected to g and h. Yeah I counted like if I take a class one graph drawing, I will flunk j. Now, two connects to f i b a a b f; I do not I am not going to continue all this. Then let us draw this, a b c d e f g. So, edges a b and c are connected to a, b is connected to a f, c is connected to f i g and so on. So, that is how the graph is connected. Then, you have one thing on and everything is connected to that off.

So, this is called a lattices structure. This way you show what is incident to what and that will also become useful. Let us alone see. So, I have not to be very particular that which presentation I use. It depends on the application and the algorithm would be that it should be able to use, generate whichever the presentation you need. So, the question is, if I give you set of n lines and what does it mean? So, each l i is given as let us say, an equation y is called ia x plus b i. So, in order to present a line, I just need to give you the coefficients. Am I right and you to give a values of a i’s and b i’s.

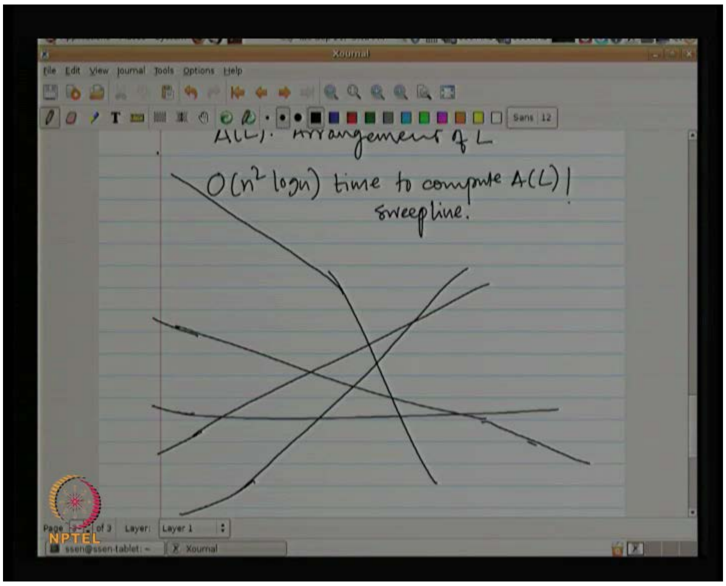
So, what you think about is that what is given to you is the set of n pairs a i’s b i’s and given that information, you want to, you just construct. Thus, you want to construct the arrangement and by constructing the arrangement, I mean if you construct the d c e l of this planer graph or you construct this lattices structure. Now, among the algorithms, you have seen there is anything occurs to you that you might be able to use to construct this arrangement. Pardon, no in this class, so because as how will you construct this quickly.

The spectral, how you are going to use the spectral algorithm? So, you learnt how to compute the intersection points of segments. Am I right? So, how do you compute the intersection because here are full lines? So, how do you compute the intersection points of segments lines? So, how much time does a line will take? N log n plus n plus I log n. Am I right?

So, what is i here n squares. So, how much time will it take? n square log n. So, compute by using a street plan algorithm? You can do it n square log n and there you just computing intersection points, but by doing additional work, you can also compute with the latest structure. Because of every which a vertex you can update the structure, you can construct with this and I will not go through that detail.

So, question is that we know that we can compute it. So, I should say if one and let me also use this notation. I will use al to denote the arrangement. Let me write it. So, what we know is, we can compute it in time n square using sweep line. Now, the question is, can you do better? So, what is a lower bound n square because the output that I want to construct as n square.

(Refer Slide Time: 37:30)

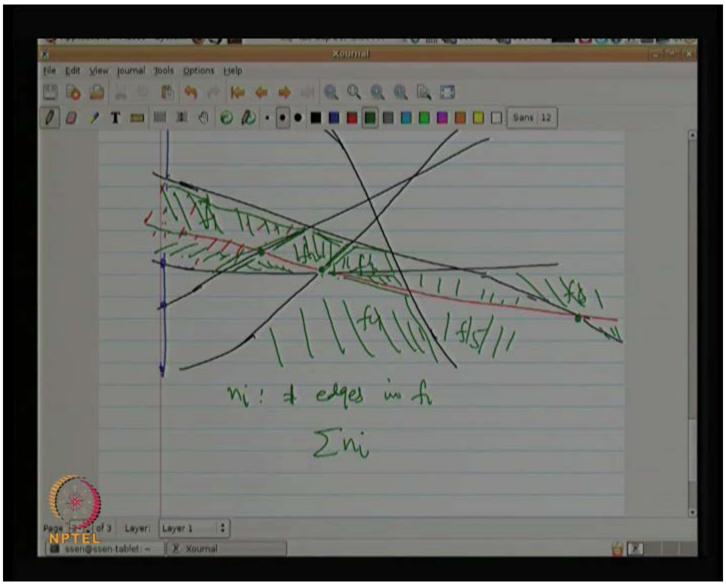


So, question is, can I do it n square type instead of n square log n? Can I get it to log n factor? So, here is a very novel algorithm to construct the arrangement. What I will do is I will use an instrumental algorithm to construct the arrangement. I will add one line, one after another. So, I will add a line. So, far let us say the invariant will be added i lines, and I have maintained the arrangement of the first i lines where I pull out a new line, and then see update the arrangements. Let us see how I do it.

So, suppose this I have done so far and it to draw new line. So, think about, let us make the first observation, the following before I do that. So, if you look beyond to the left of x is equal to minus infinity. Basically, minus infinity is hard to imagine. I am guessing about it to the left of any intersection points. So, if I draw vertical line here which is nothing to the input line and when I look at the lines, they order in which intersect. So, if invert order they appear yes. So, if I go from the bottom to up at the bottom, I will have the line with the highest slope and at the top, I have the line with the lowest slope and they sorted by the slope.

So, I can maintain this sorted list. Am I right? I will basically, what I will do, I will maintain the sorted list there what the lines I have added, so far sorted by the slopes. That takes only n log n time and they bring it, can do that. Now, when I add a new line, then I know where exactly it starts because I know it is a slope. Now, because I started with the slope, I know its position. So, suppose I am, let us say I am writing this line, so what I will do is, I will trace this line, and think about conceptuary tracing at line. It is from left to right and I will update the graph as I go on. Now, what happens? I know where it starts.

(Refer Slide Time: 41:00)



So, I know that it starts here. So, I know the face that it contains, in which it lies. Now, when I add this line, what happens, this face gets split into two. Now, what I will do is, I will take start from, let us say the top line that will lie above it on this z line and I will start following this edge. So, I will start following this edge, because I have… Let us say, d c e l structure, some structure. Then I can do that, so first I check this edge and I check with the red line intersects this edge. If I do then it is going to create a new intersection point there, and is going to move out to the next face. So, when I check it, I find it does not intersect.

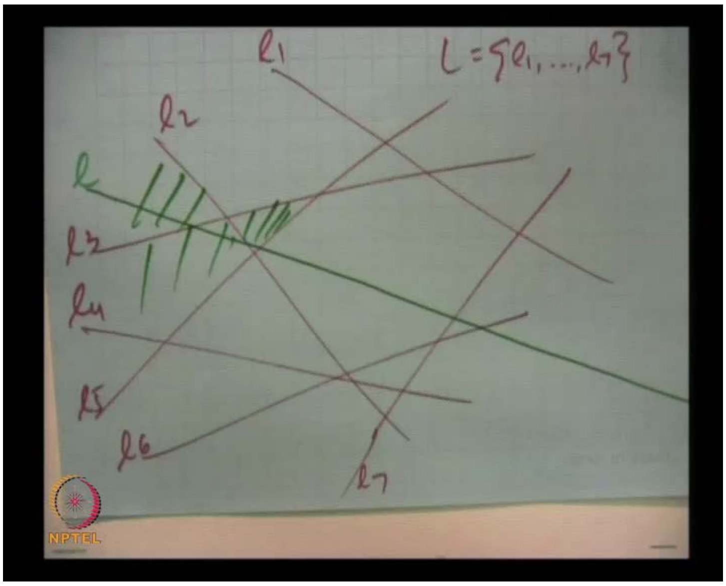
So, the intersection point I checked the next line now edge you have that intersects. It does it indeed intersect. I will generate this intersection point, create a new vertex. This edge gets split into two edges now. So, I split these two edges and also I will split this face. Now, I know that because this is a new edge. So, now what happens is this face I split into two. Now, I know that which is the next face it goes to because I know which edge intersect it and for each edge, I also know what the other faces.

So, now, I start on this face again. I start traversing this face while it enters through this vertex. I will go, let us go to the next edge. I see that there are dead line intersects these edges. It does not. So, I will come to the next one. It does indeed intersect here. I create again, do the same thing. It any vertex it split that edge; it split this face into team. Then, again I go to the next face and so on. Then it comes here again. I will start traversing this face and then I will do the same thing here. So, question is how much time did I spend on traversing this line?

That is right. So, let we start of this. So, what is happening is that the total time is spent when the line crosses the ten faces, for each face I traverse that face. So, time spent on each face is proportion to the number of edges. So, let us say these were the faces f1 f2 f3 f4 f5 f6. These are the faces that intersected and let us say an1 is a n i is a number of edges in a f i. When the total time is spent is roughly this much. Am I right? Because this is a total line spent. Now, the question I have to ask is what is this quantity? Now, you can easily have it. You cannot do a simple local argument, because you can have the situations where face can be pretty large. It may have all n edges and a line will intersect n faces.

If I do a nine analysis say I intersect n faces, each faces has n edges. So, total time spent is n square. So, intersecting a single line, this can take n square time and I am inserting n lines one after another. So, that will give you an cubic algorithm. So, we went backward. We want it the goal was in prove from n square log on to n square, but it went to n cube, but they are something to come to a rescue as if you notice that. Let us take this example. Suppose, if a line I intersect that one face has very large complexity but if you look at the nearby all the other faces, they are small faces. What all going to argue to you is next let when you do a global counting argument, and look at all the faces the dead line intersects.

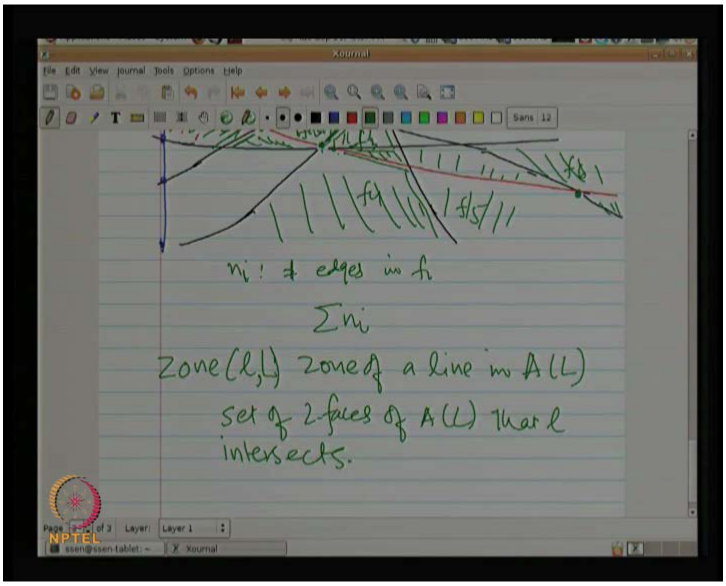
(Refer Slide Time: 46:17)



Even though a single face may have a linear n complexity, but the total number of edges and all the face at line intersects. That is also is only order of n. So, let me start that and prove that for you. So, for that I need to define some terms. So, here is a call, is definition and which I am not done it yet.

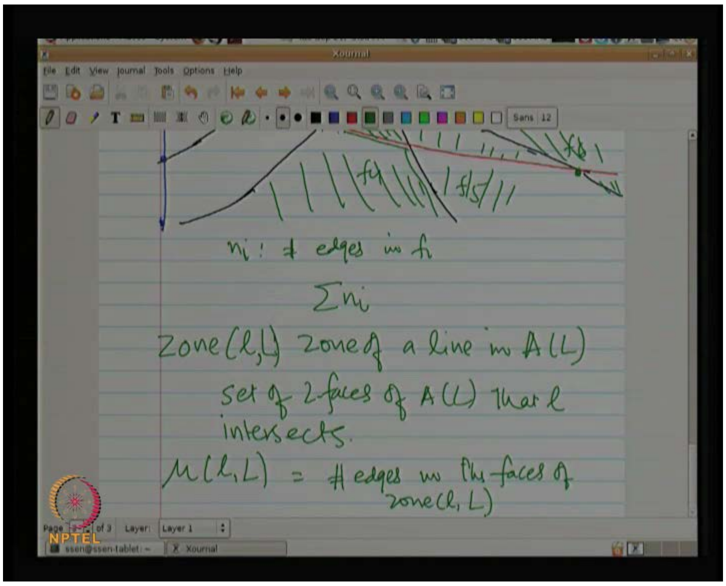
You have a set of lines. You look at the arrangement and you have some other line. So, in this picture, l is capital. l is a set of red lines you are looking at the arrangement and this is a . So, zone is a set of two faces of al that l intersects. So, this is the set. So, if you think about where the sets of faces that the line intersects. Other way of thinking about is a set of the region of the arrangement. The face of arrangement that the line can see you think about all the red lines are optic.

(Refer Slide Time: 48:16)



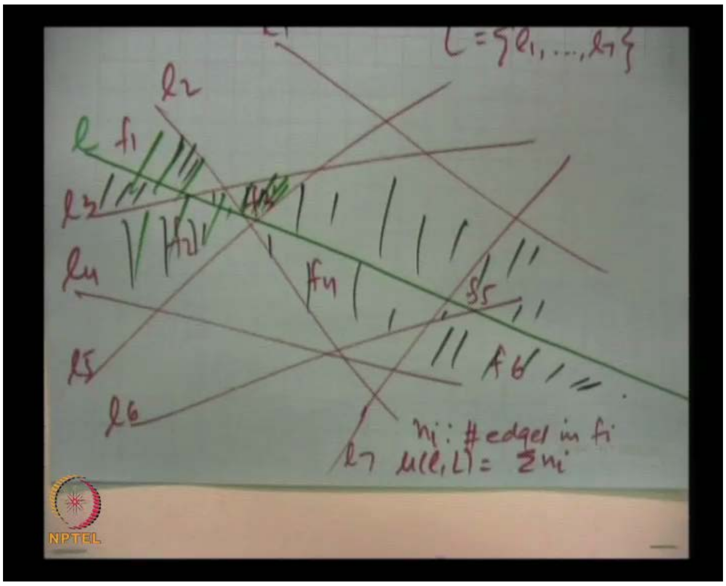
So, you cannot see beyond the red lines. So, question is, which portion the red lines arrangement you see? This is the faces that discreet the line in green line intersects and that is precise what is you sort of see. Then, I want to introduce the term called new ln is the number of edges in the faces of. So, this is a new l l is a for each of the face in the zone. You count the number of edges in each of the faces and that you sum it up. That is a zone.

(Refer Slide Time: 49:27)



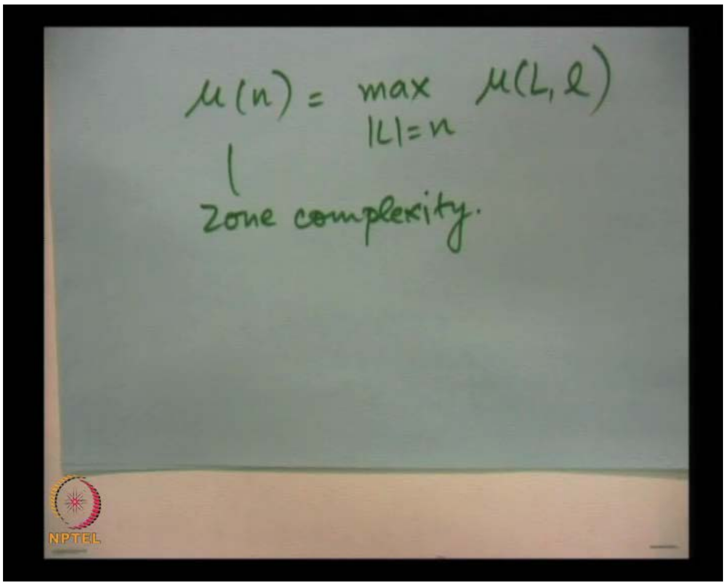
So, if you let us say this is f1 f2 f3 f4 f5 f6. These are set of face in the zone and let us, as I said ni is an number of edges in f i then mu l l is a sum of n i.

(Refer Slide Time: 50:04)



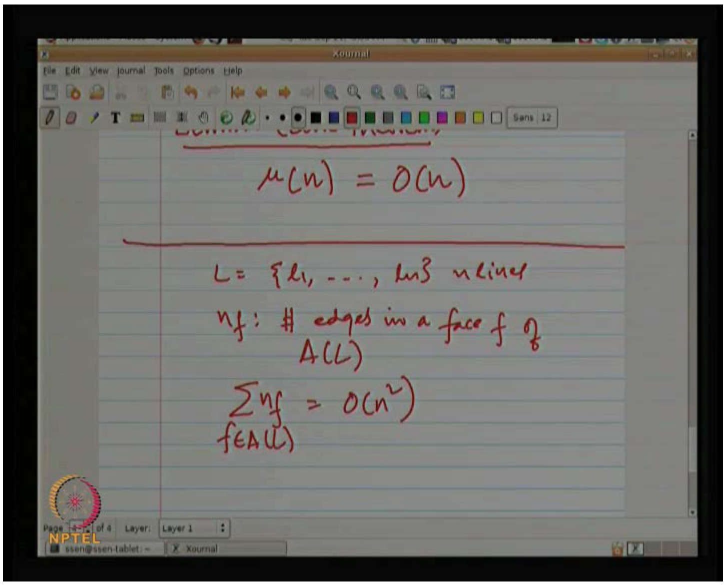
So, if I go back, then when talk about what is a running time of the algorithm, the running time of time spent in adding a line is nothing but this quantity new i, because this is the number of edges in the face that line intersects on. The claim is which is actually called zone theorem.

(Refer Slide Time: 51:34)



Let me do one more quantity. It will be easier to talk about it before I say zone theorem. Let me write it here. So, let us say mu n is the maximum. So, what is it out says is what is the maximum size of a zone can be and mu this is called mu l l is called. Thus, this is called the zone complexity. What is the complexity of a zone in the worst case? The claim is that the complexity of the zone is a new i. So, what I will do is, I will prove this theorem in the next class, but in the next five minutes, let me prove that one very interesting consequence of this zone theorem. Then I will do the proof in the next class.

(Refer Slide Time: 53:30)



So, let me go back. So, what we argued is that the total number of edges in the arrangement is quadratic n square. So, for each face if you sum up the number of faces, number of edges and sum it over all the faces if the quantity is quadratic, so let me write it.

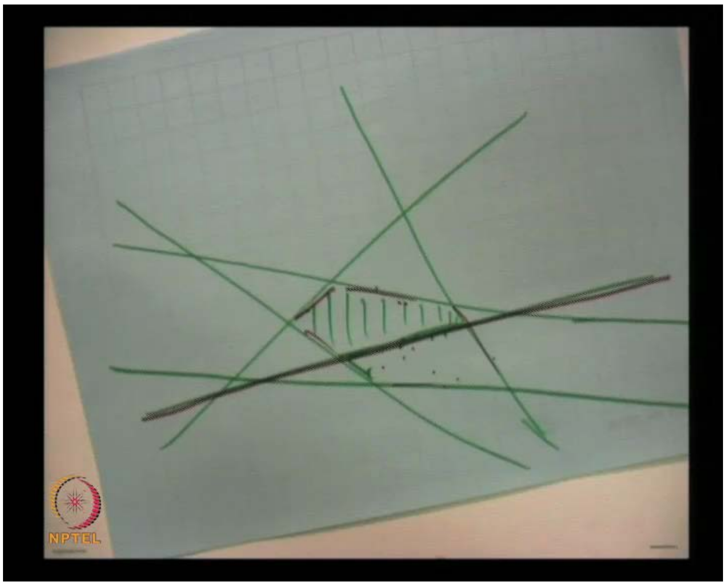
So, we will give a different exercise. So, n f is the number of edges in a faces. What we know is if you sum this quantity over all faces, it is over n square, because if you notice in the arrangement, basically what is happening is that you count each edge only twice. It is not because each edge belongs two faces the number of edges and squares. So that when you sum the number of edges in each face if the quantity is n square.

Now, we also know that the complexity of a single face can be n square. So, n the complexity of now what is amazing is s the following, but not only a sum of n f is order of n square, but if you take the sum for each face look at the number of edges take it is square and sum it over that is also n square. To appreciate that why this is surprising is a following from that arrangement has n square faces. We know that the complexity of one face can be as large as n.

So, it means that for a single face quantity of n f square can be n square. So, the n i bound on this quantity will be because there are n square faces and for single face n f square can be n square. So, ni bound will be n to power 4 and what this says is fine that some faces can be as big as n, but in very strong sense, the average size of a face is only the constant. Even the previous all set suggest that even this one was as I said because the number of faces n square and since, the total number of edges in faces n square that of suggest that but this is a strongest treatment that not only the sum of the sizes is n square, but the sum of a squares of the self size is also n square.

Now, in order to appreciate full episode give you and write. This is just a simple algebraic manipulation. Now, we know that the sum of n f over all the faces is n square. So, this is n square. The second term leads only n square. So, only thing what I need to do is n choose f and here is a geometric argument. Let me give a geometric interpretation of this first term and the last quality.

(Refer Slide Time: 56:10)

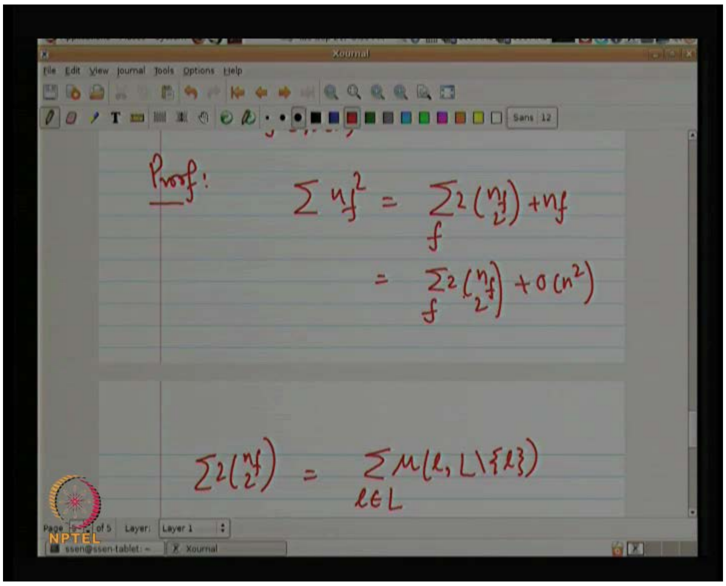


So, let us look at what is n f? n f choose to mean if you fix a cell and it is twice the number of pairs of edges. Am I right? So, for if you fix a line here, when if this one you are pairing with this one as I say this one. Similarly, so this line is also incident on the other side. So, you are also looking pairing this one with this one, this one with this one, and this one with this one.

Now, if I mark this line as a special line and I look at a zone complexity when you count the zone complexity for this cell. So, think about the arrangement and remove this red line that I have drawn. Look at the remaining lines and what I am counted. Look at the zone of this red line and the remaining lines. Then for this what I am counting is the number of these edges.

So, what I am doing is in some cells, I am counting a number of pairs for these lines. I repeat this exercise for each line. One may then think about, you pull this line, think this as a special line. Look at it is a zone in the arrangement of this of the remaining lines now.

(Refer Slide Time: 59:09)



So, what I am doing is this nf chose 2. So, chose 2 is nothing but is this quantity. It is a zone of all the l. For each line, it looks its zone in the remaining lines and that we do it. This you know is where the zone theorem is order of edge. So, this is on square and this twice is at one thing I swept through it is twice is there, because each pair is being counted twice. If you fix a pair let say this line and this line pair once we are counting when we taking the zone with respect to this line. Other time, you are counting when we taking the zone with respect to this line that remaining line. So, each one is being counted twice. That is what happens? So, that is why we get n square, so let me stop here.