### AL\_week4\_lecture\_1-20240923

说话人1 00:00  
The one that just the height of the stable. Anyway, today we are going to study a new topic, and this topic is called online algorithms. Now, we will introduce the teacher online very, very soon, but we'll start with a simple example. This example is called c rental problems or this c rental. Imagine you are doing a location somewhere in the north and there are some mountains with snow on it. Then you can see there. The good news is that you don't have a limit on your location nets. So basically, you can stay there for, as long as you like. But there is one condition. The season of the scheme, you don't know how long it could be 1 day, at least 1 week, it could be 1 month. So you don't know. All right. So basically in the evening or maybe in the beginning of every day, you will know whether you can continue skiing or not.

So now the question is, you love skiing and you would like to ski as long as it is allowed. Okay? And then you can choose between two options. One is you buy a set of c you can buy ac set, you can buy it, or you can rent it every day. All right. Actually, the decision is something like in the morning of every day, you decided to buy ac set or to rent ac set. Okay? Then right now, buying this c set will be more expensive than renting. So here we use $10. Buying is $10, renting is $1. Now, which way would you choose? Buy or rent? It depends on how long the season is, right? But the problem is don't know how long the season is. So you can only do the best you can, right?

So in that case, the trouble can be formulated in the following way. So basically, the duration of the sea season, we don't know, right? Then, as I said, in the morning of the day, you need to decide whether you buy the c set or you read the c set, right? Here we use the value m to represent the price for buying, and then value one for price of renting. And then, again, on the morning of every day, you will know whether you can continue another day or you have to go back, because it's not possible that you see for sunday and then rest for sunday where you see is possible.

And then see again, that's not possible, right? This season of the ski is somehow continues. So it says continuous segment. All right. Now, if you already buy the already bought the ski set, then you can reuse it anytime. All right. Basically, if you make a decision to buy it on the first day, then you no longer need to worry about any further actions. Okay? Now, this is the problem. Now your goal is to spend as little money as possible, right? During the whole season. Now, if the length of the ski season by t days, then the optimal strategy is very easy, right? Everyone knows? T if the number of days t is larger than m for every cent, right? You get five. This is the best strategy. If it's less than or equal to m like this, they will just register again. Right? This is quite straightforward. So we have two cases to consider, and then you have the best strategy towards any one of them, right?

This offline problem is also called offline optimal search. But the offline was the whole input, but you do not know the whole input, right? You don't know how long the season will last. So you have to go for the online setting. Now, the onset thing, you only know the information piece by piece, right? And then at every round or on each day, you need to make a decision, say, buy or rent. But you cannot say today I iii buy the c set, and tomorrow we realize that the season ends then have a chance to make please twice. Right? I changed yesterday decision to be rent. Is it possible? That's not cannot do that, right? You cannot revoke your decision on each day. All right. So the decision of buy or rent, you cannot report. But the problem then is to make a decision every day, whether you buy or you rent. Now they cost ranking one, buying m and you have t days in total, but you don't know the t okay?

Now, actually, a there are two very easy strategy, which I already mentioned. The first strategy will buy 1 %, right? Don't care about whether you have a very short season or not anywhere by first aid, by the first aid. This strategy sometimes good, especially when the season is very long, right? But when the season is short, it's not good. For example, if the number of days, t one, and then the optimal solution is ready for one day. Ready? One day. The cost is 10 k but you spend the cost of the end. We use the opt to represent the optimal cost. Aoe is the online area cost. You see the optimal solution, the cost of one, but your solution because you buy this key set, the solution for the ascent, right? To send any cost, any dollars, the ratio is very bad. Right?

Now, ii didn't formally introduce this ratio yet, but you can use knowledge in approximation equation.

For now, ok you need to compare your algorithm and optimal solution. My ratio is everything of divided by optimal solution is m m is pretty bad. This is very bad because m can be arbitrary, large. Right? This strategy is not good. How about second strategy? Sentence really says, ii never thought I just keep mentioning. Right? This is another extreme. So in this extreme, what do you see? Keep renting? It's very good if the season is very short, right? But how about if the season is very long? For example, this cheese far larger than them. And then your optimal solution, the quality is, because optimal solution will just buy ac set on the first day, right? Ok passes then, and your solution because it keeps wraps every day. The total cost is t number of days in the season. If you divide these two values, it is at over m but you see this t can be larger than m by a very good extent.

It can be 2345. Any background t can be one, beta, and m it's also possible, but maybe it's not that possible, because we have a 365 days, but probably there's a place where the cc will never act, right? It holds on every year, right? Every day you'll never end. You will then stop seeing. And in that case, this situation, t ranch every day is very bad, right? The ratio will be very bad. Ok again, this strategy is not good. Now, as I told, you, could use the so called approximation ratio to understand the performance of the algorithm, the online algorithm compared to the optimal solution. But in our scenario, we need to introduce another issue now for approximation issue, because there is some kind of underlying difference between these two models, approximation algorithm and online algorithm.

Let me define the trouble first. Here we are talking about, you want to spend as little money as possible to finish the whole season, right? Ok we define the problem as a minimization problem. We try to minimize certain costs. Here, your algorithm has an output right algorithm, I and then your optimal solution also has some output. Then we divide algorithm output by optimal solutions output. This ratio, if for any possible problem instance, it is larger than equal to one, or rather than equal to alpha.

Actually, let me see this max value. This max value is the ratio of your average. Right? This guy will be larger than one in minimization problem, that a legal call, this mass algorithm over optimal, the competitive ratio for development. Here it is a different word, competitive reason. Okay. Then they asked me, why would you not put approximation issue? Remember that approximation ratio is for what kind of problems it's hot problems, especially on behalf of you cannot do the best solution in reasonable time. You sacrifice quality of solution and say, I want to use a polynomial time to complete a relatively good solution.

Now, because of my constraint on computation, power, I have to both for approximation power, I have to live with the approximation.

Now, but here it's a different story. Why? Because you don't have efficiency problem, I guess you see what's your decision. The decision is. You see today whether if you can still see or not, and then you make a decision I buy or rent, right? Do you have an initial issue? No, right. It's too easy. It's just finally choice, right? And also really based on your previous knowledge, and maybe remember 1 bit of information of the previous decision, right? And then you have the current decision for today. You don't have any issue about computation. Right? Even you use it very, you may need a mobile phone, very old style without you can still do it, right? It's very easy.

The difficulty of the problem does not come, because the problem is difference. It's because of all here why you can get optimal solution. It's not because of computation power limitation, right? It's about or your current decision will affect your future result, but probably we need to break future into our consideration, said probably the the difficulty of our problem comes into play because we do not know what the future. We do not know the future. I do not know tomorrow whether you can still ski or not, right? This is uncertainty in the future, which makes the problem hard, right?

So here, the bottleneck or the difficulty comes from the uncertainty from the future, not the if insufficient computational power. All right. Now, because of this, we call it competition. Now we have two strategies earlier, right? One is a violent birthday, the second one is the branch every day, right? And then the question is, these two strategies also come out that in some cases, right? Now can we have one strategy which is good? Any time no matter which case it is no matter, the season is long or short, this extremity is always good. Can we do that? Now the answer is otherwise I will not use this one as an example, right? Okay. So let's see how to do it. Now I give you one example show. It cannot be too good. All right. Now because we are trying to improve the ratio sale, if we earlier, this is really big, can we get the ratio of 2 or 1 . 5 or 1 . 1? Can we do the strategy which has a good ratio or even better ratio?

Now here i'm giving you one example to show you. This ratio cannot be too good. All right. That is not as good as 1 . 5. Right? So let's see. We use the example empty group, so which means buying is twice the cost of rent.

So in this case, now, we are trying to design something or some person called anniversary, which means a bad guy trying to make your algorithm perform bad badly. All right. You have never, I don't know what it is, but i'm trying to behave in such a way that your elder will suffer from my behavior. Or your evidence will perform very bad under my constructive input. Ok this is my goal, or also your goal. Ok you are trying to design something which will make the algorithm feel not so good. Let's see. At the very beginning, on the first day, the algorithm must have made one decision, either by, all right, according to the errors output, you're going to construct your instance, right? You are going to construct your bad example. For example, if everyone says, i'm not gonna say, and then what would you do? Now, everyone says I like it, right? What are you doing on the second day. And so it's supposed you are weather controller, you can control whether it is raining or not, also not raining, snowy or not, whether it is suitable snow or not.

Okay, so then if the album bias On the first day, then what are you going to do? Your suit? I those use for the second day or not suitable skill to see on saturday, right? So you will force the player or the person to stop. It's easier. Tomorrow, tomorrow, no more skiing. All right. So you see that in this case, you will force the second day to be not suitable to see, right? And then your algorithm already paid a cost of two, but optimal solution only to rent for one day. And then finish the season. In this case. What's the ratio? Two? Back, 2 / 1 is 2. How about if the algorithm decides to rent one person? And then you say, because the events I cannot stop the season tomorrow, because if I stop, this area will be optimal. Right? So I do not stop to see that. I let you continue your seat on saturday. So then please, I will make your decision down, right?

So again, I was gonna say, how about why on the second day, and then we'll leave it. Stopped on saturday, right? You will never stop, right? After the algorithm bias.

Now, in this case, what's the comparison? The comparison is, you pay 1 + 2, right? Three costs, but optimal solution only need to pay. Well, good. The ratio is one 1 . 5, right? In this case, another 1 . 5. If you rent on the second day, this, like you read on the second day, and then again, you will not stop, because if you stop this algorithm, again, it comes out, right? And there's some good. I keep the season on. All. Right? And then on the survey, the algorithm, again, has two choices, buy or rent. And then if you buy, again, I you will stop the season, right? As long as somebody buy has another device. If you stop the season, maybe this case, the album pays 1 + 1 + 2, right? 1124, and then the optimal solution.

Again, we need to pay $2 because you can buy on the first day, ok in this case, four divider two, the ratio is two. Right? So one last chance, rent keep renting ok so now I do not need to go on being aaa nice person letting you continue to ski, because now you already accumulated cost 123, right? You have cost me. So it's very safe for me to stop it on the 4th day, because even I do this, I stop the season.

Now you already pay $3. But optimal solution only need to pay one to us. 3121 . 5, right? Ok so you see, I have enforced all the possibility of errors for any so called deterministic algorithms. This is the only possibility, right? By event. Ii allow you to believe in the environment. I allow you to continue the environment. Okay? Done. I I stopped the instance, right? In all the possibility or all the possible branches, the ratio it happens, 1 . 5. So that means you cannot do, or you cannot design an algorithm whose of a competitive ratio is better than 1 . 5. Good. Now, at least we know we cannot do better than 1 . 5. So can you do? 1 . 5? Maybe not. Let me or let us try some a bit conservative thing. Let's try to prove we can do ratio two. All right. So can we do it too?

Now, let me remind you one more time. The harness of the problem does not come from communication. It comes from what? Uncertainty or lack of information, right? You don't have sufficient information. That's why you cannot design or get the optimal search ok now let's see, can we do two about. All right.

Just now we have two strategies. One is, if on the first day, second is you keep renting, but actually, there is series of mechanisms or strategies, which is something like. Because but after buying, after purchasing, after buying this, he said you do a lot in red, right? This is something for sure, because after buying, they use it whatever. All right? That means your last section. This guy. But before you buy it, you can choose some days to rent. The general strategy looks like what? You? First rent for some days, right? And then what? Five. So basically, all the strategies will look at this. All right. So let's see how to do it. It's very important five after. And thanks. This is a very, very special script. This is one of the strategy sets. I told you just now. This is a special combination of bible, purple, renting for some days in denmark, ok now I set the threshold to begin. You will know very soon why I set the threshold to the end.

Now this is the picture, day one to dm you rent, right? Day one to day m you rent. Then it's suffered am plus one. You buy virus, right? So what's the total cost to them? If the season is long, this is the end, right? If the season is short, very good. Let's see why it is very good. Now, if season is short, which means t is animals m then actually you are doing what optimal solution is doing, right? If season short, optimal solution is never buy, just keep renting, this is what optimal will do, right? Ok so, in other words, in case one, you are doing as good as optimal solution.

Now, when the thing is long, if you are not often, but not too bad, why not too bad? Because when the season is long, your optimal solution will buy, right? Will buy at first aid and use it forever, right? But you're also not too bad, because you you only pay a cost of what? 200. This is the maximum cost you have to pay back. Your cost is to them. Optimal solution cost is m right? What's the ratio? Ok good. You already have the two competitive areas, most eventual problems, right? Is it right? Okay. Now we are about to use the paper when we do use the paper during the break, but we are starting to give bonus points.

Now, this is a good strategy which is too competitive, right? Okay? Now, the theorem, I guess I don't need elaborate further, because the analysis is very straightforward, right? Just have a different case. And you can see that it's a two competitive algorithm. I will skip. This part is very straightforward. Now we'll see earlier, we proved that you cannot do better than 1 . 5. And now we can have a strategy whose ratio is two.

Now, a very natural question is, can you do better than two? If you can do better than two? Then you need to design a new strategy, right? If you sell, it's not possible to be better than two, then you need to find a way to truth. No one or no strategy can beat the racial right now. Which direction are we trying? Actually, we are trying to is the best possible. Let's see. This is a zero. We are trying to show. Remember the brilliant picture, this one, we can only show 1 . 5. Now, let's look at a more systematic analysis.

Okay? Basically, just now we already know all the strategies will look like this. Why? After x days of rent, okay? By rental x base, and then by all the strategy will have this kind of formats. Okay? Let's see. Suppose it is a page one. What is page one? Page one says your x is, at most, mu minus one. Right? If x is, at most, n minus one, then how do we analyze the performance? Or how do we set the adversary? If you buy after x days of rent for this strategy, then because remember, we are trying to be a bad guy. We are trying to design some sequence. Not good for everything, right? So basically, what you will do is you will stop the season, right? After five, right? Good. So now you see, I will just make the season x plus one, which means because you buy on day x plus one, right? You rent for x eight and then buy one x plus one day, right? Ok I just let you see the mean x plus one.

In this case, let's see, what's the optimal solution? You just keep reading everything, because you remember, this text is less than equal to m minus one, which means a shortly short duration matter, ok this is optimal solution, which is keep rent any day, will not buy ok so this optimization, what's your solution? Your solution is just x plus n because you rent x states.

And then five ok let's compare these two values, x plus n divided by x plus one. What is that? Now we change it to be this one plus m minus one over x plus one. Is that right? This one is, okay. Now remember, what is next? Xm os, 10 - 1. This value is at least two minus one over n you have another n coming out. Then inside is minus one over nok this value will approach two when n is super large. Right? This is one case. The second case, you'll see this wrong. Now, again, your industry back example will stop the season on day, x plus one. Right? So after you make the buy decision of, I will stop this set t as usual, to be x plus one. And then optimal solution is just bias cc set. And your solution is x plus n right? Because you run for x eight and then buy, right? This step, divide the ratio, divide the two values, x plus n over n because x is values m this value is at least two.

By now, we have proved that the ratio cannot be better than two for a significant amount of gap. For example, you cannot do two minus 2 - 0 . 001 competitive issue. Can I do that? Because I can make any 1 million also. I can make them 1 million, and then that's the values thousand and 20. Right? Ok so basically, you cannot be the ratio to.

So now comes the first for this question. Why? In previous picture, we don't need to about 1 . 5 not two. It seems that the strategy we use are very similar, right? Just stop the season after you buy. What's the reason that in the previous page only show a low down of 1 . 5? Because the previous city, he then when the x larger than the . of view, that when the rather is that this reason yeah, actually, it is not a difficult point. It is a very easy point of the value of those three or less and the value of what? Value of what? So basically, I will not go back to the slide, because if I go back, then everyone will see what's the problem.

And for the track, the previous tree structure, only prove a lower. I'm fine. Because what I just need sitting in the back, right? Because some because what's that something? I believe someone can get performance for sure. This they are very good, because you see, what's the previous page? We assume m is equal to two. When m is equal to two, when you look at this picture, what can you do? M is two is just 1 . 5, right? You can only show 1 . 5. The main bottleneck here is that you set m equal to two in the previous picture. So do you agree? This is not a difficult bonus. So I hope that the next more challenging policies together. All right. Okay. Now, so much about the ratio based on the ratio, it's just two. You can achieve two. On the other hand, you cannot do that, right? So basically, this problem, it comes to an end. Okay? This is only a reference page of others, even the problem, but that's not a major focus of our course. Let's just summarize what we have discussed so far.

Now, all my algorithm was the goal. The goal is to study some optimization problem without knowing the input in the us. Okay? And then you are comparing your algorithms output with the optimal offline error. Now, optimal flying algorithm is somebody who knows the whole language, right? So we can get the best solution for sure. You are trying to be close to the optimal solution. Now, here, your algorithm will generate some solution, which is not exactly often. There is a gap between your algorithms output. The optimal solution? No. Why do you have such a gap? The gap comes because that's right, because you don't know some future information. Okay? It's not only about the traditional power ok that's also why we do not call it approximation factor. We call it competitive vision.

Now, for different problems, sometimes we are the maximization problem, sometimes we do minimization problem. But no matter which one you do, a ratio closer to one will always be good, right?

For example, if I have a ratio is 1 . 1, it's better than 1 . 3. If my ratio is 0 . 9, it is better than 0 . 7 closer to one ratio. That will always be something you want more. All right. So, all right. Now, I think that's for now, until now, you have known the major difference between online, our family and prosecution effort, and online algorithm, the gap between the performance, and you have an optimal solution is from the uncertainty or unknown information. That for approximation ratio, the gap comes from your insufficient computing power. All right? Okay? But today, other than scheme at a problem, we are also going to introduce some other problems, including the graph algorithms, like matching, and then scheduling problems. So these two are the remaining two problems we're going to talk about in the first part. And then there are also other combinatorial management problem, which can can do online analysis, but they will not be covered in the lecture notes.

Then in the second part, we'll talk about something related to probability. Now, that includes former secretary problem, profit inequality, camera box, et cetera. Now, here we are basically talking about online secretary and profit inequality. These two the analysis is very interesting. Now, the last one camera box, you all of the name of that this side, no camera. What it is about, I already forgot. So I hope that you can use your results to find out what it is. Now, I still have a rough memory about this problem. It's basically about because camera box, right? Yes. So basically, before we open it, you do not know what it is inside. Something like that. So the problem is also similar to that flavor. You have a bunch of boxes, and then you can open one, but not sure which one to hold it. Okay? You need to make a wise decision to make you helpful better than you expected or as good as possible, something like that.

If you search for this word on the internet later on with a break, you will find what this problem is, but it's better not to do it during this lecture, because this problem is also somehow related to the profit and quality we are talking about, although not directly related, it has a similar nature.

Now, these are the launch .. You are going to cover in this lecture. And other than that, actually, online algorithms are used a lot in the daily life, including online dispatch, which means you have some orders, right? Say, actually this is about the you buy food, take away food from some small restaurants, and there are some people, some riders sending the order, sending your food to your apartment or dormitory.

So there are some kind of service like that. When different people order, they need to somehow give the driver the deliverer some orders such that he or she will take these orders and then go to abcd and send the orders to these customers.

All right? So it's online because no one knows that out who will order take with food, right?

That's some decision to make, right? And then there are also something called ride sharing, especially when you're going to somewhere taking a taxi or taking a captain car. Right? So sometimes there's optional shared car, right? The 2 groups will share the same car, and the car will pick up the people from from a right? And then b and then go to a's destination and then go to b's destination, right? Something like that. So you need to have some way to combine the orders together, right? Two or three orders together so that it will be most efficient. So this is another on my hands on my problem. And then there's also online learning, which is more abstract. Okay? Data comes one by one, then also there is a multi banded problem. So it's, again, aa very classic problem, basically meaning you put it, impose harm, and then there will be some expected war coming to you. But when you decide whether you pull the same arm repeatedly, or you go to another arm, so that the reward could be higher possible.

So this is also a very big class of problems in online center setting. Okay? But research in these topics, usually they work here, the like econ cs conferences, theory conference, di conference, machine learning conference, et cetera. So they said possible values for publication.

Now, let's look at the second example online. Bipartite mention. Okay. So what is matching or matching is actually very easy. Now, suppose that you have 2 groups of people, one group on the left, they are applicants. There are multiple jobs. All right? And then the second group is the jobs. All right? 3 months. Now, if all the jobs and packets are already there, then it's very easy to do magic. You try to match as many people as possible to jobs. Right? So basically, you will try to find, for example, such a matching where all the red advocates are matched to some job on the right. This one, you also call it the maximum matching. Okay? Is maximum possible.

Now, what is online setting? The online setting says, now I no longer know the whole graph. Instead, I want you to agreement the jobs. The jobs are already on different websites, right? All the jobs are there, but applicants do you know what applicants are? I don't know, right? Or the companies posting the jobs, they also do not know which applicant will come to apply, right? Advocates are somehow coming here, one by one like this. For example, somebody comes, one applicant comes. And then when this applicant comes, it will choose some jobs to apply for ok for example, I apply for three jobs, which are the three edges right? Connected to three notes.

So after this candidate applied to three jobs, and then imagine that and all three jobs thinks it's okay, right? This candidate is okay, then your platform to decide which job this candidate should be matched to, right? It should go to, for example, sar maybe I assign the applicant to the middle job.

So this finished your decision for the round where u comes, right? When u comes, you will make a decision. Where do you match u two, right? Okay, so here I mentioned the second one. And then there is another one coming in like this. Okay? The second one be prime. It can match to second and third and 4th jobs, but because the second one has already taken. So this one can only either be matched here or matched here. Okay? Again, you make a decision which note the second applicant will be matched. Okay? You just keep going. We try to maximize the total number of advocates matched to some job. Remember, one job can only have one applicant matched it, right? This is constrained. Construe the match. Again, because it is an online algorithm, you cannot revoke your early decision. For example, you already matched the second note to you. You cannot say because you crime comes. I will match this note to you prime, and I will choose another part of you or you're gonna do it, right? This is gonna be better.

Okay. So now, this kind of matching scenario actually happens a lot in real life, including the scenario we talk about just now about the job hunting, right? And then you can also have the online advertising where it was a match, the web pages to the potential ads on web page, right? Okay. Now, for this matching problem would have a simple, really area, but actually, I do not quite agree that this is a video. Let's see what this element is. Okay? So there it says, whenever a new law comes, I match it to one of the neighbors. This is not connected. One of the available names take an arbitrary, available edge, match it. I didn't see any media, but let's just pretend it's a video, right? A it's an algorithm breaking ties in an arbitrary way. Ok you have different choices. I just choose one of them, ok that's the algorithm. That wasn't very, very easy, right?

Now one example, here, you see, there are six notes on the right. There are jobs, right? And then your epic the task. The first one I used to match the second job, right? And the second one comes only one hash, so you cannot match. The third one comes against the neighbors. I take the middle one, and then the 4th one comes again with one edge. But that job is already taken. So you can imagine, right? And then next one come in, you match to the second job. And then last one comes in. It does not match anything. That's the way you take it. This is the red, three red edges. Are the output of your everything, right? But what's your output usage? There are six very good. The optimal solution is matched all the six horizontal path. The atom is three. Optimal solution is six, very likely. There is a ratio very similar to here.

But because we are talking about the maximization version, the ratio is not . 2, instead of the . 0 . 5, which is 1 / 2, 1 / 2. Because whatever has less matrix, often has more edges. A smaller number below that big number is 0 . 5 of transmission. Okay? This is the maximization problem, by the definition of the competitive ratio is just a mean, right? Minimum of the ratio for all the instances. Right? So this is a a competition of this of this scenario ok now this ratio definitely will be at most one, because it's a maximization problem, right?

Now we are trying to show this 0 . 5 possible 0 . 5 competitive ratio. Now, what can we do? Now, basically, we have one important lemma, show you that whenever what edge, uv is chosen by an algorithm, then and those two edges in the optimal solution will be sacrificed.

What does it mean on how that it means that the choice made by everything will broken or will submit two edges from optimal being chosen. Right? Why? For example, when you look at the red edge, right? Sorry. Every red edge here, it touches only two edges from optimal solution, right? And then so choosing the red one means that you can no longer choose the two yellow on there. All right? So that's the high level idea. So basically, it says your evidence choice will hurt. It's two edges from organization. So on the high level, whenever you choose one edge by the algorithm, you sacrifice two edges from the optimal solution. 142142. In the end, if you choose k edges, you only sacrifice two k edges from optimal solution, meaning that optimal solution might probably only have average p edges. You ever. K is pretty good, because k developed to take is one of the two, right? Now, this is not the 100 % rigorous proof, but it's the rigorous proof is similar to this. We'll talk about it very soon, but we'll talk about in the second hour, but before we end this session, i'd like to raise the bonus question, second bonus question.

This kind of argument, did you ever see it before in the approximation level? I similar arguments, similar high level argument. If you see it, where has it? For which problem, which method we use a similar idea? Comfort groups. It depends on the set of rules in different rules, but then similar comparable rules. Close, but it's the other one. So I hope, you know, what I mean by is the other one. What's the other? Who said the tenta? Vertex color. The problem is vertex color is the opposite of e and seven, right? That which method, which guy would be the best happen? Because it will test when we talk about two elements. This vertex cover is 0 . 5, who can tell me the method of that textile.

Another half. Somehow, you need to describe everything. What does 10 hours look like? The next hour it will be running algorithm. That area is more advanced, right? Although it also shows the two approximation, it's not the the one which is comparable to this one, because you see this one is too simple. And this one is very simple. What's that everything? There are only two. It is not running. Then what's that? We are talking about what? Excellent not set up. A second, we are looking for another, the other method for the text cover. What's the other method? Raise your hand for. That's what I have to go very far. Ok is that so you already give back to the what you did last week. I guess I overrun for this bonus, by the way, if no one will try the animal.

Say the answer, please. People after we were tested with them with one uncover, it is incident, too. Is it this method take an arbitrary vertex, which has one uncovered edges incident to it, but that one does not have a good constitution ratio. Take over, take both standpoints of one uncovered edge, right? That's the algorithm. Remember, you take an edge which is not compatible and take both end points, false vertices. The argument there is what? Because optimal solution needs to take one of them, at least, right? But you just need to, you just take double what optimal will do. Right? It's very good. So let's deal with it, right? Similar, not the same, but a similar argument. As this one says, whenever you take one edge, you just sacrifice the most two edges in our solution. The other one says, I take one edge, both end points. I just take twice the ultimate, because ultimately, to take one angle, I take two. Not a big loss. Right? So I guess we'll take 10 minutes break, and then for those about bonus, please write down the names. Some.