PROFESSOR: OK let's get started. I once taught with a professor who was lamenting the fact that as the term progresses attendance in lecture tends to drop off. And gets pretty dramatic by the end of the term when you're lecturing, and nobody's there. And I asked him what he did about it. And he thought about it and he said, there's only two things that can get students to come to lecture, candy and sex.

Now we've already tried candy, so today we're going to talk about sex. In fact we're going to use graph theory to address a decades old debate concerning the relative promiscuity of men versus women. Now graphs are incredibly useful structures in computer science, and we're going to be studying them for the next five or six lectures. They come up in all sorts of applications, scheduling, optimization, communications, the design and analysis of algorithms. In fact next week, you're going to see how to Stanford graduate students became gazillionaires because they use graph theoretic in a clever way.

But let's talk about sex. The issue that we're going to address today is one of the most talked about, and most well studied, questions in all of human sociology. On average, who has more opposite gender partners, men or women? Now opposite gender is going to be important. And by this I mean, one boy, and one girl. All right, I'm not making a political statement. It's just that the math is a lot easier that way, as you'll see.

Now I'd like to start by taking a pole here to see what you think about that. So raise your and if you think men, on average, have more opposite gender partners than women do. Only a few.

AUDIENCE: In life or [INAUDIBLE]

PROFESSOR: Um, you can--

[LAUGHTER]

PROFESSOR: One on one. OK, so let's say over the course of their lives, let's say, or over the course of 2010, that men in America have more opposite-gender partners than women in America, say

in 2010. Raise your hand if you think men have more going on. All right a bunch of you. Raise your hand if you think women have more opposite-gender partners? This is unusual. Maybe even more voted for women, but it's close.

Raise your hand if you think it's equal. All right, about the same. Raise your hand if you think there's no way to know, that it's hopeless to really figure it out. All right, nobody goes for that. All right, good.

All right well now in the popular literature, I think the feelings are different than expressed here. Pretty much universally, in the literature, it's believed that men have more opposite-gender partners than women. And in fact, you could even think about that, if you think about literature, the leader of the harem is always a man. And he's got lots of women. In polygamist cultures, it's always the man that has multiple wives, not the reverse.

Now not surprisingly, this issue has been studied "scientifically," I'll put in quotes, extensively, in one of the largest studies ever done. Researchers from University of Chicago interviewed 2,500 people, at random, over several years. They brought them in, on many occasions, to try to get the answer for the question once and for all. And they wrote this 700 page book, called The soul of *Social Organization of Sexuality*: *Sexual Practices in the US*.

Actually walking around with this book has proved to be a little embarrassing. Last week my 11-year-old daughter saw it, and she goes dad, why do you have this sex book. And I grabbed it back and said, well that's for the course. I'm teaching. And I thought I'd gotten away with it, and everything was fine. And then later that day she texted all of our friends about the new news that what do you know, her dad teaches sex ed at MIT.

Anyway this study concludes that on average men have 74% more opposite-gender partners than women. There's one other central claims.

And this is in the US.

OK now, when you think about it that sounds maybe reasonable, might be OK. But not according to ABC News. They did a poll of 1,500 people in the country, in 2004, and concluded that the average disparity is much greater. In particular, in this study, they said that the average man has 20 partners-- I'm assuming over their lifetime-- and the average woman has six. And this gives a disparity 233%.

So ABC News did a smaller survey says that it's 233% here, much more than 74%. Now ABC News claimed this is one of the most scientific studies ever done. And there was a 2.5% margin of error. Now we'll actually talk about what that means mathematically later in the term when we do probability, and do study polling. Now of course I should also mention that ABC News is the one that said Al Gore won the presidential election in 2000.

Now the study is called American Sex Survey, a Peak Between the Sheets. That doesn't sound so scientific. And it was on TV, on Primetime Live in 2004. The promo for this is really good. It says, a groundbreaking ABC News Primetime Live survey finds a range of eye popping sexual activities, fantasies, and attitudes in this country, confirming some conventional wisdom, exploding some myths, and venturing where few scientific surveys have gone before. By the end of today, we're going to agree with that last statement.

OK now who do you think's right? University of Chicago. Who votes for 74% as being pretty close? A few of you. I've already slammed these guys. Who votes for ABC News as being more accurate? Yeah, nobody. Who votes for no way to tell? I got some votes there, all right. So how do you tackle this problem?

In theory we could do our own 6.042 survey. I don't know how much we'd really learn, and for sure I'd get fired. So I don't think we're going to do that. But fortunately, this is the kind of question that could be handled, and actually answered, by graph theory, even though it might be more interesting to interview thousands of people, and find out what's going on. That's not as efficient as using graphs.

So let me start by defining what a graph is. Informally graph is just a bunch of dots and lines connecting the dots, it's actually very simple. So here's to graph. These are the nodes, and they're connected with these lines, called edges. And often the nodes, and sometimes the edges, are labeled. For example, we might call this x1, x2, x3, x4, x5, x6, and x7. So that's an example of a graph.

Now this being a math class, we got to give a formal definition of a graph. And we'll usually use the formal definition. A graph G is a pair of sets often called V and E. Where V is a set of elements called vertices or nodes. And it has to be non-empty here in this class.

And we'll go back and forth between vertices and nodes. Even the text we use both words

interchangeably. And E is a set of 2-item subsets and V, and they're called edges.

So for example, over here in this picture, V is the set of nodes is x1, x2, x3, up to x7, that's the nodes. And E, the set of edges, is pairs, unordered pairs of vertices. So for example x1, x2 is an edge. And it's the same as the set x2, x1, doesn't matter the order here. Later in a week or so, we'll talk about directed graphs where the order matters. x1, x3 is also an edge here, and so one. Think we've got, let's see, 1, 2, 3, 4, 5, 6, 7 edges in this graph. And the last one would be x5, x7.

Edges are also sometimes written with this notation, x1 line x2, is another notation. And then later when you talk about directed edges, we'll put a little arrowhead on one end of this.

Now the definition of a graph is really pretty simple. Just think of it as dots and lines, if you want. But there's often differences in how people define graphs. For example, in this class we don't allow the empty graph, i.e. the graph with no nodes. So we're going to insist that every graph has to have at least one node in it. And that's just to make the theorems we're going prove be true. Otherwise there's some theorems that are false for the special case of the empty graph.

But we don't require the graph to have any edges. In fact, it's possible you have a graph with nodes, but no edges. For example, this graph. Three-node graph. So here G equals VE, V equals x1, x2, x3. And E is just the empty set. Now for a general graph, when you do have edges, we say that two nodes, call them xi and xj, are adjacent if they're connected by an edge, namely if xi xj is an edge.

All right so for example, x5 is adjacent to x7, but it's not adjacent to x4, there's no edge there. Closely related is the definition of the incidence. An edge E, which is xixj, is said to be incident to its end points, xi and xj. OK so, for example, if I labeled that edge as E, E is the edge x1, x2, and this incident to x1, and incident x2.

Then we can talk about the degree of a node. The number of edges incident to a node is called the degree of the node.

So for example, what's the degree of x5 over here? 3, so in this case, the degree of x5 equals 3. The degree of x7 is 1. These guys all have degree 0, there's no edges incident to them.

Now in this class, we're going to look at only simple graphs, at lease for a while. A graph is

simple if it has no loops, or multiple edges. Now a loop is an edge that only connects up one node, that's a loop and we don't allow it. A multiple edge is we've got two edges that are really the same, they connect the same endpoints. Also called a multi-edge. And those we're not going to have in simple graphs. We don't allow this. We don't allow that. Any questions so far about what a graph is?

So how are we going to use a graph to model the problem of opposite-gender partners? That's the question we're after. So any thoughts about what the nodes of the graph are going to represent? What is it?

AUDIENCE: Males and females?

PROFESSOR: People. Yeah, so we're going to have people. In fact, there's two kinds of people here. There's men, and women. All right we got nodes here for the men. And in fact in America, there's a lot of nodes here. All right, and so this might be oh I don't know, say that's Tom Cruise and Nicole Kidman. Now what's the edge going to represent?

AUDIENCE: Partners.

PROFESSOR: Partners. They were opposite-gender partners. And there's actually more edges probably here. We could have Penelope here, and Katie here. And well probably lots more, I probably don't know them all. And Ben's over here with Nicole. And Nicole got Jude and Keith. There's actually a website you can go to get a lot of these things here. And Katie went with Josh. It's called whosedatedwho.com, and you get big graph, you could start filling in the edges. I don't know how reliable it is.

Now it's really critical that we're only looking edges from here to here. All right, so if there's an edge between Tom and Ben, I don't want to know about it. Just opposite-gender partners. OK now in the USA, the number of nodes here is about 300 million. About three million people. And the number of men nodes, male nodes, call these VM, and this is VW, by the way, I'm using cardinality notation. When I put bars around a set, that is the denoting how many are in the set.

In the US there's about 147.6 men out of the 300. And the number of women-- oh we got a w here-- is about 152.4 million. So there's a little bit more nodes on this side of the graph, than that side in the US.

What about the edges? Any idea of how many edges there are here? We don't know. I sure as

heck don't know how many edges there are. So that we don't know. The cardinality of the edge set we don't know, and we're not likely to figure out. I don't even think these surveys, really, can estimate that. But what we're trying to figure out is the ratio of the average degree of the men, to the average degree of the women. Because the number of opposite-gender partners you have is your degree here, and you're looking for the average guy degree, compared to the average female degree here. That's what we're after. All right so let's find that quantity.

Let's let A sub m equal the average number of opposite-gender partners for men. And we can let A W be the same thing for women.

All right. Now we're trying to figure out the answer to this question. What is A m, the average guy degree, over the average woman degree. And in particular, the University of Chicago says, they say it's 1.74. That the average guy as 74% more opposite-gender partners than the average woman. ABC News says it's 3.33, that is 233% more for the men, than the women. Now we're going to figure this out what this ratio is. Just use a little bit of math here, and a little bit of graph theory.

So let's write a formula for A m. Well we're trying to figure out the average degree over here. Well, that's pretty simple. We just add up all the degrees, and divide by the number of nodes. And that'll give us the average degree. So the average degree is the sum of the degrees, over all men, x in the set of men, of the degree of x, divided by the number of men. Can somebody give me a simpler expression for this? It doesn't have that nasty sum in it?

AUDIENCE: E.

PROFESSOR: E. The cardinality of E. I'm adding all the degrees here. Well that's just another way of counting all the edges, because every edge shows up once, and only once, in a degree count here. And this is where, we use the fact we have opposite-gender partners. Because if I had some edges over here they wouldn't get counted in sum of the degrees here. All right so this is just the cardinality of the number of edges, divided by the number of men. Any questions about that? Because this is an important statement about graphs in general.

When I have a graph like this-- which is called a bipartite graph, we'll talk about more in a little bit. But where the edges go from the left to the right if I sum the degrees on the left, I'm just counting the number of edges. All right, let's figure out a formula for the average number of

partners for the women. That simple that's just sum x over the women. The degree of x, divided by the number of women. Let me rewrite that so it's clearer. What's a simpler expression for this?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yeah, this sum, adding the degrees of the women, is just the number of edges, right. So that is cardinality of edges, divided by the number of women. All right, well now we can write, solve for our formula, average over men over average of the women. That's E over VM, divided by E over VW. Wow, this is nice. I don't know the number of edges is, but it just canceled out. And this is just the number of women, divided by the number of men.

And in fact we know that. That's this number, divided by that number, which is about 1.0325. So we just proved, that on average, a man has 3%, or 3 and 1/4% more opposite-gender partners than women. No need to do the interviews, or spend years doing. That is the answer. And it has nothing to do with the promiscuity of men, or women, nothing at all.

So the Chicago study is way off, and the ABC New study is completely nuts. It just can't be right, this is a proof. Now what happened here? Well what's going on, what's the reason for why this is true? Yeah?

AUDIENCE: A male has a female partner then the female has a male partner.

PROFESSOR: Yeah.

AUDIENCE: You're not looking at like how many males are going to one female. The promiscuity isn't even a part of the question.

PROFESSOR: That's right. It takes two to tango. Every time you got a guy, you got a women. And you have the number of relationships going. The average for the men is that number, divided by the men. Average for the women is that same number, divided by the women. And so if there's more women, they're going to have less partners on average. Has to be. So it really was a stupid question. It's very, very simple to answer.

Now as it turns out there are endless studies like this, in the literature. In fact, a few years ago the Boston Globe ran an explosive story about the study habits of students on Boston-area campuses. And their surveys show that, on average, minority students tended to study with non-minority students more than the other way around.

And they want on great length consulting the experts as to why this might be true. Why is it the minority students study with non-minority students more than the other way around. Now can anyone tell me why it is certainly true, and not surprising, why that's the case?

AUDIENCE: Because they're the minority.

PROFESSOR: Because they're a minority. There's fewer minorities than non-minorities. End of story, we don't need this sociology PhD from down the street to explain it to us.

We're going to see a lot of other bogus studies later. This is not unusual, especially when we get the probability. Just every day there's a new one in probability. Any questions about this before we leave? Unfortunately that's most all we'll say about sex today.

OK. But now, in this example, we used an edge in the graph to denote some kind of affinity between two nodes. The two nodes liked each other in some sense if they were connected by an edge, or they had a relationship of some kind. There's lots of examples in computer science where you use an edge to denote just the opposite. That the two nodes can't be near each other, or don't like each other.

For example, consider the problem of scheduling final exams at MIT. And they do this after they find out all of your schedules, and they try to schedule the exams so that you don't have to take two at once, or there's as little of that as possible. For example, let's do an example here.

Say we look at these five classes. Take 6041. And this may not be totally accurate, but roughly. So I've got five MIT classes, and I'm going to put an edge between pairs of classes that have overlapping student enrollment.

So in this case, for example, we've assumed in the drawing of his graph, that you can't have our exam the same time is 6002, on the assumption there's students in both classes. But you could have our exam the same time as 6034. Because there's not an overlapping student in both classes, so the exams could be scheduled at the same time. So we've used a graph to represent which courses can't have their exam at the same time.

Now let's also suppose we have a set of slots for the exam. And say they're all on a Wednesday. And the first slot is Wednesday from 5:00 to 7:00. And the next one is 7:00 to 9:00. And then, the next one is 9:00 to 11:00. And then 11:00 to 1:00 in the morning, and then

1:00 to 3:00, getting pretty late. And your job is to figure out how not to have to use these later exam slots. You'd like to use as few as possible so you're not going too late night, or come before the holidays, so you're not having exams on Christmas and New Year's, for example.

So the goal is to assign slots to the nodes. Put every node in a slot so you don't have nodes hooked by an edge getting the same slot. Now this is an example of what's called a graph coloring problem. So let's define that.

Given a graph G, and K colors, assign a color to each node, so that adjacent nodes get different colors. All right, and then the minimum number of colors you need is called the chromatic number of the graph. So the minimum value of K, for which such a coloring exist, is the chromatic number OF the graph. And it's denoted by this symbol chi of G. Because usually you want to use a small number of colors.

Now what does a color represent when we're dealing with this problem? What's the meaning of a color?

AUDIENCE: Time slot.

PROFESSOR: A time slot, OK. So let's call this time slot C1, C2, C3, C4, C5, so there's five possible colors. Now of course, we could color this graph with five colors, every node could just get its own color. But then somebody's taking their exam from 1:00 to 3:00 AM, and that's a bit of a pain.

Let's see if we can do less than five. Let's say I give this color one, let's give this one color one, that's OK, because they're not connected. I can't give this one color one, so I give it color two, say. Now this one I can't give color one, because this guy got it, he can't get color two, because that guy got it. So it give it color three. And well, I can't do one, two, or three here, so I gotta go to color four.

All right so 6042 will get the 11:00 PM to 1:00 AM slot, not so good. Can we do any better? Can we get away with three colors. Some say yes, some say no. How many people think you can do three colors on this graph? A bunch. How many think you can't do any better? All right, the vote is mostly for three. Let's see. Any ideas? Anybody see how to do three? Yeah?

AUDIENCE: Assign C4 to 6034 .

PROFESSOR: Assign C4 to 6043.

AUDIENCE: Or C1 to 6042.

PROFESSOR: C-- I can't do see C1 to 6042. It crashes, but can I do-- yeah? Put

AUDIENCE: C1 in 6003.

PROFESSOR: C1 in 6003.

AUDIENCE: And get rid of C1 in 6034.

PROFESSOR: Get rid of--

AUDIENCE: Make it C2.

PROFESSOR: Make this a C2. Oh, yeah. All right, these got C1, they're not adjacent. These got C2, they're not adjacent. This can now get C3. So we can have our exam from 9:00 to 11:00, which is better. All right, can anybody do it in two colors? Can anybody offer a reason why two colors may not be possible? Yeah?

AUDIENCE: Because let's say you could do it with two colors.

PROFESSOR: Yep.

AUDIENCE: 6041 and 6002 have to be different colors.

PROFESSOR: Yes.

AUDIENCE: 6042 can't be C1, and it can't be C2.

PROFESSOR: Yeah, good. So you can't in two colors, because these three guys would violate that. You've got a triangle here. Each one of these guys has to be different than the other two. So two colors can't work. You've got to have at least three in this case. So three is optimal. We have just shown for this graph, the chromatic number is three.

All right, now in general doing what we just did is very hard. No one knows a fast algorithm for determining the chromatic number. In fact, it's a weird kind of problem, because it's easy enough to check that a coloring is OK. If somebody put a coloring on the board, you can check, oh that works really simply. Just check every edge, and make sure the colors are different.

But figuring it out, as best we know, you've got to try an exponential number of possibilities. So

if I had 100 nodes here, my running time of the algorithm to check all the possibilities would be exponential and a hundred. Yeah?

AUDIENCE: Can that number just like the highest degree of each node, or nodes.

PROFESSOR: Uh no. But it's no worse than something like that, as we'll see a few minutes. That's a great observation. And we're going to come back to that in a few minutes. But it's not just that.

OK now in fact even figuring out for an arbitrary graph if three colors can be done, called the three-coloring problem, that's really hard. No one knows how to solve that in less than exponential time. In fact, one of these NP-complete problems is what it's called. How many people here don't know about NP-completeness? Is everybody-- all right so all of you haven't seen NP-completeness.

OK so there is a class of thousands of problems-- in fact there's books list these 1,000 problems-- that are all NP-complete, somebody's proved they belong in the class. And what that means is that if somebody gave you a solution, like a coloring here, it's easy to check really quickly if it's valid. But figuring it out is really hard. And if you figured out how to solve one of those thousands of problems, like suddenly you figured out how to tell if any graph could work with three colors, you would solve automatically all other thousands in the book.

So it's this book of problems you will constantly run into in your career in computer science. And it's bad when you run into one, because there's no good algorithm to solve it known. But if you just solved one of them, the other thousands would suddenly be solvable quickly. Even better, you win a million dollar prize. One of these Millennium Prizes we talked about the first lecture.

Even if you show you can't find a fast algorithm for one of them, that means that known of them have fast algorithms, and you also get a million dollars. So this is the central problem in computer science, and theory computing, is whether or not you could solve these NP-complete problems.

Now actually lots of people have claim to do it. And in fact, there was a lot of buzz in the community about a month ago when actually a reputable researcher at HP Labs said he'd done it. He proved that you can't solve NP-complete problems. And he got people going for probably at least a week, until they discovered a fatal flaw. And the proof was actually bogus. So no one still knows if you can solve these NP-complete problems quickly.

Now the problem is, in practice, you run into these things all the time, like MIT really does have to schedule the exams. So you've got to do something. You can't just go say, hey it's NP-complete, so no exams this year, or whatever. That's not going to fly, so you got to do something.

So now this is a problem-- many of you when you go into careers, you're going to be faced with this. You got to do something. Any thoughts about an algorithm for coloring graphs that might use a small number of colors? It doesn't have to always work, or you're going to win a lot of money if it does. But a simple algorithm, you can't take either the 100 steps. You got to be linear, probably, or quadratic time. That could get you a small number of colors. Any thoughts about what you'd do? Yeah?

AUDIENCE: The number of degrees and nodes?

PROFESSOR: The number-- what about it?

AUDIENCE: The highest degree and that node, the 6042 is [INAUDIBLE].

PROFESSOR: Yeah.

AUDIENCE: So you could use that.

PROFESSOR: Good, all right. So what do I do with that-- so I found a node with a high degree, there's three of them have degree three here. What do I do with them?

AUDIENCE: Pick a different color to.

PROFESSOR: Pick a different color, that means I've colored some of the others. If I pick a different color, do I start with them, or do I finish with a high degree nodes? Because you've got to assign the colors to them. And high degree is important to be thinking about. We're going to prove a theorem in just a minute about related to degree and coloring.

AUDIENCE: Start with them.

PROFESSOR: Start with them, and do what with it? Color?

AUDIENCE: Yeah, and then assign the ones that aren't connected [INAUDIBLE] to the same slots.

PROFESSOR: OK, so I could-- here's a degree of theory now I can start with color one for that. And then

what do I do next? I pick-- its neighbors have to get different colors, I guess. You'd start coloring the neighbors.

AUDIENCE: My first instinct would be to color all the [INAUDIBLE].

PROFESSOR: OK. And what color would use for them?

AUDIENCE: Different ones.

PROFESSOR: Different ones if they're connected, or if they're not connected you'd still use different ones?

AUDIENCE: Only if they're connected.

PROFESSOR: Only they're connected use different ones. And so if they're not connected, you'd use the same colors? Yeah? You're going close, and it actually works pretty well. The underlying principle you're sort of thinking about here is you've got some notion of the order in which you're going to process your graph. And you're going to start with a high degree nodes, in your case. And as you go along, you're going to start coloring the nodes. And you're going to make sure you color them legally. And it sounds like you're going to color them with a low color as you go along.

And that is probably the most basic graph coloring approach. And almost you could almost say is a generic approach. So let's define that, and then see prove some facts about it.

Most of the graph coloring algorithms in practice are based on this approach. And we're going to call it the basic graph coloring algorithm. And for our graph G, with vertices V, and edges E.

So the first step is going to be to order the nodes from 1 to n. Now in your case, you were suggesting an ordering where I have the high degree nodes first. All right. But for now we're not going to specify that. We're going to make it any ordering you want. And then we're going to have a notion of an order on the colors, as well. And I don't know how many colors, but they're going to be numbered 1, 2, and so forth.

And then we're going to process the nodes one at a time, to N. We color the nodes, what is step I, we color the Ith node V sub i with the lowest legal color. And by the legal I mean you don't color at the same node as another node that's already been colored the same that it's adjacent to.

All right so let's try this. In fact, this is sort of the algorithm I used initially to color exam graph over there. All right, so let's look at that.

So let's say we-- let me erase the colors here, and put an ordering on the nodes. So let's say I ordered them with 6034 first, so this would be V1. Then 6041 is V2. Then V3, V4, V5. If that's my ordering, what color would I assign to 6034?

AUDIENCE: One.

PROFESSOR: One, C1, I'd color it first to get C1. What color does 6041 get? C1, as well, it's the lowest possible color that's legal, and is not hooked to this guy, so C1 is legal. What color do I give here? C2. Then I color this one next C-- can't do C2, can't do C1, so I pick C3. And then I get to 6042 last, and I can't do one, two, or three, so I do four.

All right so algorithm, with that ordering, gave four colors. However we know there's a way to do a different ordering that gives us three colors. In particular, let's see if we do this what happens if we use this other ordering. Let me erase these.

Say that's V1, V2, V3, V4, V5. Now I get C1, this will be C2, C1. What's this one get? C2. Ah, much better. C3. So different orderings result in different numbers of colors here. So the whole art now becomes finding a clever ordering. And so many people have already had good ideas, pick the largest degree nodes first.

And in fact, if you simulate the algorithm on lots of graphs, you do better on average when you color the larger degree nodes first. And then if you start to use more exotic orderings, you can do even better. If you take a lot of graphs that are out there, and run your algorithm, and see how well you do, you do better with more sophisticated orderings.

In fact, this was my senior thesis back when I was undergraduate student. I was trying to figure out better and better orderings that worked for graphs. And at the time it caused a bit of a problem. I was a undergraduate at Princeton. And Princeton, to this day I think, still has exams after the holidays, the Christmas holidays, New Year's holidays. And the students wanted to have the exams before Christmas, because they hated going home for the holiday, and then you've got to worry about your exams when you come back. And the faculty said no, there's no way to get them all compressed into a small number of days.

Now I wasn't aware of all that of the time. But my thesis was go figure out good ordering. So I tried lots of different orderings. And I tried the largest degree first, and recursive versions of

that actually worked very well. And then tried it on the Princeton exam graph. And lo and behold, you could actually squish it down, so you could give all the exams, I think was, 4 and 1/2 days, plenty of time to give them before Christmas. Which caused a fair of scandal at the time, because then the faculty had to come clean that they just didn't want to bother having the exams before Christmas.

Now this algorithm is an example of what's known as a greedy algorithm. Now in a greedy algorithm it's always simple. You just go one step after the next, taking the best you can do at each stop. You never go back and try to make things better. You never do hill climbing, if you're familiar with that term. You just always keep it simple, one thing after the next, very fast. Sometimes it works great in practice. Sometimes it doesn't. But it's always where you start, some simple approach like this.

Now this algorithm actually, even if you don't try to monkey with the ordering, even for a worst case ordering of the nodes, that actually does pretty good for a lot of graphs. And in fact, it does really well-- as somebody already asked about-- if all the nodes have low degree. So let's state that as a theorem. And then we're going to prove that.

So if every node in a graph G has degree, at most, d-- so that's the biggest degree in the graph, D-- then this basic algorithm uses, at most, d plus 1 colors for G. No matter what the ordering is, you'll never do worse than d plus 1 colors.

So what's the value of d for our exam graph over here? d is 3. Every node has degree, at most, three. And so it says, that no matter what ordering you picked here, you'd get at most four colors. Now you might do better. In fact, we found an ordering that got three. So it's possible to do better.

So let's prove this fact because this makes a difference. Say you have a graph with hundreds of nodes. But every node has degree, at most, three. Well that says you only need four colors even, if the graph has 1,000 nodes, and that's very useful. So in that kind of situation it does very well. So let's prove that. Any ideas as to what proof technique we're going to use?

AUDIENCE: Invariant.

PROFESSOR: Invariant, close. Not quite an invariant, but close.

AUDIENCE: [INAUDIBLE]

PROFESSOR: What?

AUDIENCE: Well ordering principle.

PROFESSOR: You know well ordering principle, yeah, we're going to use the equivalent version of that. We're going to use induction. If you like well-- it's equivalent to well ordering. If you like well ordering you could do it that way. I think it's easier using induction here. So the proof is by induction.

All right so the first thing we need is an induction hypothesis. Any thoughts about what the induction hypothesis should be? Yeah?

AUDIENCE: If you have a graph with n nodes then where the degree of any nodes is less than [INAUDIBLE] then you can do it.

PROFESSOR: That's great. You're going to do really well on the midterm, because you put an n into this thing, but there's not an n here to start. What are most people going to do-- we used to ask this actually. We asked this once on a test many years ago, and it was an utter disaster, because did everybody do? May be one student, or two, put an n into there. But what's the naturally thing to do to induct on here when you look at this statement? You're going to induct on d, because the first thing you do is you make this be your induction hypothesis. There's only one thing to use, so you're going to have your predicate be p of d, and it's going to be that.

Now It didn't occur to us that's what everybody was going to do, but it should have. They all did that and it was a disaster. Because if you do this, well you've got to take a graph with maximum degree d, or d plus 1 in the inductive step, pull out all the nodes with degree d plus 1 to get a graph with now degree d. And that's a mess. You just pulled out a lot of nodes, potentially. Color that in d plus 1 colors, now put all that junk back in. And say only used one more color. Nightmare. And these were MIT students under pressure. It was a nightmare.

So that does not work. And in fact, we will ask an induction question on graphs on every test you take in this course. It will happen. And so usually, with induction, you take this as your induction hypothesis. With graphs, you have to be careful. And worst part about this is we tell people when this doesn't work, use a stronger induction hypothesis. So students tried to make a stronger, but they're still stuck on d, and it was still a disaster.

With graphs, you do something different. And the first thing you do with a graph, usually, is put n in here. And if it doesn't work with n, the number of nodes, you put in e the number of edges. And induct on that. And so what you said is exactly the right thing to do. Don't do this, or least don't spend too much time on it. Pretty quickly try this. If every end node graph-- if every node in an n node graph G has degree at most degree, then the basic algorithm uses at most d, plus one colors. And now you induct on n. And almost always on graphs, that's the first thing to try. Even if it's not in your theorem statement. Any questions about that?

Well let's start with this, and see if we can make this one work. So what's the next step in our proof? What do we got to do? Base case. And the base case will be, not n equals 0, because we can't have a zero node graph, but n equals 1. And how many edges do we have? Zero. If there's one node, we don't allow loops, so it's zero edges, which means that the degree of our graph has to be zero. There's no edges. And of course there's only one node, so one color is going to work, and that happens to equal d plus 1.

All right, so the base case is true. For one node graphs, you can always use d plus 1 colors, where d is the max degree.

All right, next we have the inductive step. So here we assume P n is true for the induction. And now we look at an n plus 1 node graph to show P n plus 1 is true. So we let G be any N plus 1 node graph. We got to show you can color it in d plus 1 colors. And let's let d be the max degree, the largest degree in G.

We've got to show we can color it in d plus 1 colors. Well the basic algorithm, let's say. First thing we do is we order the nodes in an arbitrary order. And we're going to show whatever order you pick is OK.

All right so what are the nodes? Anyway at all. Now how am I going to use the induction hypothesis? I know, I can assume, the for any N node graph I can color it in the max degree plus 1 colors. How am I going to use that to help me color G here, the n plus 1 node graph? Any thoughts? Yeah?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yeah, let's create an n node graph by looking at these nodes, and taking this one out of the time being. Remove the last V n plus 1 node in the order. That leaves an n node graph. So let's write that down.

We remove the n plus 1 from G. And that creates a new graph, call it G prime with vertices, V prime and edges, E prime. So we create a new graph by removing that node. And we remove all the edges tied to that node.

So for example over here, the last node was 6042, so we take out 6042, and all these edges. And this is a graph that we're left with. That graph has n nodes. What's the maximum degree in G prime? When I pull out a node, can the degree of any node go up? No, I'm just taking stuff out.

So I know that G prime has maximum degree, at most, d. The degree didn't go up of any node. Might have gone down, but it didn't go up. So G prime has max degree, at most, d, and it has n nodes. So we can use the induction hypothesis P n. It says that the basic algorithm uses d plus 1, at most, d plus 1 colors for nodes V1 to V n. Any questions about that?

So if this were the n plus first node, last node in the ordering take it out. The basic algorithm now, take the same order here, V1, V2, V3, V4, basic, we'll color that in d plus 1 colors. And all I have left is to give this guy color, and I'll have color G. Question? No.

All right. So by induction I've colored these guys, V1 to V2, and d plus 1 colors, all that I have left to do is color V n plus 1. And hopefully we're not going to use color d plus 2, because then we sort of-- it wouldn't work. We got to use one of the first d plus 1.

All right, so let's look at V n plus 1. And let's call its neighbors in G, U1, U2, Ud. It has, at most d neighbors, because every node in G has, at most, degree d. A neighbor's a node you're adjacent to.

All right so, V n plus 1 has at most d neighbors, is adjacent to, at most, d other nodes. Now what does that mean about the color I can use on V n plus 1? What do I know about what color I can use for that? Yeah?

AUDIENCE: It can't be any of the colors of U1, U2, and so on.

PROFESSOR: It can't be any one of these colors that were assigned here. That's true. So how many colors got ruled out? At most d, and how many am I working with? d Plus 1. So I got one left that I can use safely. OK.

So this means there exists at least one color in my set of d plus 1 colors. It's not used by any

neighbor. And we're going to give V n plus 1 that color.

All right. So now I've colored every node in G, the n plus 1 node graph, safely using a total of d plus 1 colors. So that means the basic algorithm uses, at most, d plus 1 colors, on G. That means P n plus 1 is true-- whoops-- and the induction is complete. Any questions? Yeah.

AUDIENCE: Could you also start from the other way, and start 1, go to 2 nodes, 3 nodes at each step keeping all nodes at all other nodes. [INAUDIBLE]

PROFESSOR: What do you mean by keeping all nodes connected?

AUDIENCE: [INAUDIBLE] each node has an edge connecting to each other one.

PROFESSOR: OK so, then I get a specific graph. I start with this, I add a node and make it adjacent. I add a node and make it adjacent.

AUDIENCE: [INAUDIBLE]

PROFESSOR: Yeah. So you've constructed a particular graph. This is actually called, for the n nodes, it's called Kn, is the n node complete graph, also called a clique, like a clique of friends, where everybody likes everybody, in a clique.

And in fact for n here, for those n nodes, what's the max degree? Max degree is n minus 1. What's the chromatic number of this graph? What's the minimum number of colors?

[INTERPOSING VOICES]

PROFESSOR: And they all have to be different, which is d plus 1. So you have built a special graph for which the optimum of number colors is d plus 1. But that is not a proof that this is true for all graphs. Because you've looked at a particular graph here.

AUDIENCE: [INAUDIBLE]

PROFESSOR: What's that?

AUDIENCE: [INAUDIBLE] It means that you can still use your less than or equal to sign.

PROFESSOR: I see, so you'd add a node, and it's only connected to a few of them.

AUDIENCE: No, it's connected to all of them, but it still implies that you need less than or equal to the

colors. It turns out it happens to be equal to.

PROFESSOR: Yes, in this case that's right. So you've made an argument for this case where it actually is equal, but that only worked for this graph.

AUDIENCE: [INAUDIBLE] worse case.

PROFESSOR: It is the worst case, so it meets the bound. It shows you cannot improve this bound. Yeah, is there a question up there?

AUDIENCE: All I was going to say is that you've proved it's the worst case.

PROFESSOR: Right, so what you've done here is you've shown that I could not make that theorem any stronger. I could not replace it with d here. All right. Because you've given an example where I can't get d colors, where the maximum degree is d. But that doesn't-- To get a proof for a theorem, I got to go through all this. That wouldn't give me a proof of the theorem.

They're not equivalent. One's an upper bound, one's an existence of a lower bound. This shows that for any graph, you need at most d plus 1. So any graph, at most. That shows there is a graph that you need at least. And they are not equivalent.

All right. One is for all, and upper bound. The other is there exists a lower bound. So different in two ways that are important.

This kind of proof is very typical for what you'll see with induction in graphs. And you'll get a lot of practice with it. Are there any other questions on this proof? OK.

All right, see we've seen now, by that example, we can't improve the theorem. In some cases, though, the theorem is way off, for some graphs. Can anybody think of a graph where the bound we get from the theorem, of d plus 1 colors, is way off from the actual chromatic number you need, the number of colors you need? Yeah?

AUDIENCE: [INAUDIBLE]

PROFESSOR: What is it?

AUDIENCE: A graph [INAUDIBLE] two sets of [INAUDIBLE]

PROFESSOR: Good, OK. Yes, so what if we did this graph. Let me draw it out. So you've got a bunch of

nodes here, bunch of nodes here. And every node here is connected to every node over the other side. And if this is an n no graph, and I've got n over 2 on each side, what's my degree here? What's my max degree of this graph?

AUDIENCE: N over 2.

PROFESSOR: N over 2. So d is n over 2. What's the chromatic number? How many colors do I need for this? Two. All right, so d plus 1 is way off of two. There is a even worse example. Yeah?

AUDIENCE: That graph where you have one node center that's connected to a bunch of nodes regularly distributed about.

PROFESSOR: Yeah, the star graph. All right, so I got one of the center, I got n minus 1 outside. So here the maximum degree is n minus 1, just like a complete graph. But how many colors do I need? Two. So it's even worse here.

All right now what about the basic algorithm? How well does the basic algorithm do on this graph? Or to the vertices some way? Color on one [INAUDIBLE] lowest color. How many colors is it going to use?

AUDIENCE: Two.

PROFESSOR: Two. It doesn't matter the vertices. V1, V2, V3, V4, because I'll color this one 1. What am I going to call that one? 1. Then I get to the center, what am I going to color it? 2. And now all the arms, what do they get colored? They all get 1. Whatever order you pick, you get two colors.

All right so now there's a difference between the theorem just gives you an upper bound, it says, at most, d plus 1 colors. But in fact the algorithm can do a lot better than that, as on this example. So the algorithm might be a lot better. Everybody see that what we're doing here? How the algorithm is better than the bound we proved by the theorem, even though the bound was pretty good for some graphs.

Now it turns out-- I mean we're not going to win a million dollars for this algorithm. And in fact, this algorithm is sometimes very bad. And a really bad example it's very close to this. In fact actually this one, let's look at how well does basic do one this one here. Make some ordering. V1, V2, V3. What's the basic algorithm going to do on this complete-- it's called a complete bipartite graph, is what's this called. I'll define bipartite in a minute-- but what's the basic

algorithm do here? Any idea-- does it take n over 2 colors, or does it take 2? Any ideas? 2.

So take a vertex, and the first one, say V1s here, get C1. As long as I keep picking vertices over on this side, they're going to get C1. As soon as I get to a vertex over here, what color does it have to get?

AUDIENCE: C2.

PROFESSOR: C2 because it's touching the very first one we had here. So when I get vertices over here, they're all going to be C2. When I go back over here, they're going to be back to C1. So actually basic does good here too, gives you two colors. Yeah?

AUDIENCE: [INAUDIBLE]

PROFESSOR: Ah, those two aren't connected. But this case, if I've got a vertex over here it is, by definition, connected to the vertex over here. Because every possible edge is here. But that's a great idea. What if they weren't all connected, that's actually a great idea.

In fact, the nasty example for the basic algorithm is very much like that. Let's draw it. Because so far, the basic algorithm is pretty much done perfectly on all the graphs we looked at even when the theorem wasn't tight. So here is a nasty graph. And it is very close to the graph we just look like, where all the edges are there.

In this case, all the edges are there, except for the one straight across. So if this is-- the edge denotes likes, this is a world where you like everybody but your spouse. All right, so you have an edge to every one, except the one directly across from you. No edge there, and so forth. So it has almost every edge, but it's missing these edges.

Now the basic algorithm might do well here. What would be a good ordering for this graph to label these V1 through Vn? Yeah?

AUDIENCE: Go through everything on the left side, and then the right side.

PROFESSOR: Yeah, that's right. Because then color 1, color 1, color 1, all the way down. One color for the left, what does this one get? Color 2, because it's hooked up against. And these all get color 2, so I've used two colors. Really good. Basic algorithm's looking great.

Now here's a harder question. Can you figure out a bad ordering for this graph, where I use a

lot more than two colors.

AUDIENCE: [INAUDIBLE]

PROFESSOR: What is it?

AUDIENCE: It starts at the top of the cross, and then the next level then across.

PROFESSOR: Very good. V1, V2. Just as natural, really, if think about it, to order it this way.

All right. What color does V1 get? C1. What color does V2 get?

AUDIENCE: C1.

PROFESSOR: C1 because it's not hooked up here. What color does V3 get?

AUDIENCE: C2.

PROFESSOR: C2. What about V4?

AUDIENCE: C2.

PROFESSOR: C2. It's not hooked up. It can't get one, because that's up here. And it's not the two, so it gets two What color does V5 get?

AUDIENCE: C3.

PROFESSOR: C3. Because it's hooked up to one to two. V6 ?

AUDIENCE: C3.

PROFESSOR: C3, it's hooked up to one and two, but not three. And you can see what's happening here. All the way down here he's hooked up to all the n over 2 minus 1 colors. So he also takes C n over 2. So if you pick that ordering, not so good. You use n over two colors. So it really matters the ordering.

Now I should say graphs like-- actually any questions about what we did here? About this? All right, now I should say that graphs like this have a special name, they're called bipartite graphs. And that's important to remember.

All right, so a graph G is said to be bipartite if the vertices can be split into two sets, or partitioned, and we'll call them a left set, and a right set, so that all the edges connect a node in the left set, to a node in the right set. So in fact, a lot of today we've been looking at bipartite graphs, because the nodes are here. Like the men, and the women, and the edges only go from the left to the right. And that is called bipartite. And it's called bipartite because you can do it with two colors, or in two pieces.

So you don't win a million dollars for deciding whether or not a graph can be colored in two colors. That's easy. You'll even do it for homework one of these times. You do win the million dollars for deciding if a graph can be colored in three colors. That's really hard to do.

Now coloring problems come up in all sorts of applications. You know with this company, Akamai, that came out of MIT, we've talked about. We run a network of 75,000 servers. And they're used to distribute content on the internet, and so forth. And we have to deploy a new version of our software on those servers, pretty much every week. We're pushing new software out. And you can't deploy on every server at the same time, because you've got to take down a server to deploy new software on it. Got to take it out of commission.

And so we can't just take down all 75,000 servers, because then all the Facebook, and Netflix, and all those sites would stop. That would be bad. And we can't do them one at a time, because there's 75,000. And it takes a few hours for each one to get the traffic off, stop it, load new software, and turn it back on. And it would take us years to do one software install, which we got to do every week.

So we've got to figure out a schedule for how many servers you take down at a given time, and which ones. And it turns out pairs of servers have certain critical functions. So there's certain pairs of servers you can't take down at the same time.

So we have a gigantic 75,000 node coloring problem, where there's edges between servers. Nodes are servers, and there's an edge between if you can't install new software at the same time. And so when it turns out, when you run one of these graph coloring algorithms on it, you could do it with eight colors. It just turns out that way.

So that means there's eight waves of install that go on to the network. And now eight times a few hours each means that we can do it in a day, and you can manage it.

You know on a much smaller scale, the same problem exists for register allocation, for

variables. Here you've got to assign every variable to register. But you can't have variables that are active at the same time associated with the same register. And you want to minimize the number of registers you need.

So again, you have the graph coloring problem. The number of colors is the number of registers you need. And two variables can't get the same color if their active at the same time, so you put an edge between them.

The most famous example of graph coloring is the map coloring problem, with the four coloring theorem. And so here, every country is a node. Adjacent countries have an edge between them, because you don't want to color adjacent countries the same color, or you can't tell they're different countries.

Now the last example we can talk about is an important problem in communication theory, communication networks, where again coloring comes up. Now here you need to assign frequencies to radio stations, or the cell towers. It comes up in mobile networks, or just in with radio stations. And if two towers have an overlapping area, they can't be given the same frequency, so you get collisions between the towers. And frequencies are very expensive. Companies pay the government a lot of money to get certain spectrum.

So suppose you had this problem. Here's tower A, this is A's range, where it reaches. Here's tower B, so it overlaps some with A. Here's tower C. Here's tower E. And here's tower D.

All right now the question would be, how many radio frequencies do you need? What's the minimum number of frequencies you need to enable all the towers here? We could make that be a graph. There's a node for each tower. And an edge between towers, if they overlap. C doesn't overlap with B, E does. E overlaps here. And then D overlaps here.

So how many frequencies do you need for this graph?

AUDIENCE: Four.

PROFESSOR: Four would work, three is better. Can you do two? No you can't do two, because you got here. But you could do three. You could do one, two, three, two, one. This problem comes up--

AUDIENCE: [INAUDIBLE]

PROFESSOR: Did I screw up? Ooh, no I can't do that. One, two, yeah much better. All right, this problem

comes up all over the place. I'm certain you'll see it sometime in your career, you'll have some problem, or you're scheduling something, and it's really a graph problem in disguise. OK that's it for today.