In the previous video, we used linear optimization to allocate ads.

In this video, we're going to use a greedy approach to allocate ads to queries.

The approach is called greedy because we allocate the ads sequentially and we prioritize which combinations of ad and query to use, based on the average price per display.

So here we have a spreadsheet set up in the same way as the one from the previous video with some minor tweaks.

If we scroll down, we have this table here, which has cells corresponding to combinations of advertisers and queries, which tells us basically how many times we display a particular advertiser's ad with a particular query.

But to the side here, we have another table which we will use to keep track at every stage of our allocation process, how much we can allocate-- how many displays we can perform of a particular ad with a particular query, based on the budget, and based on how many displays of each query remain unallocated.

Further down, where we have our constraints, we've added some new cells here, which are labeled with the word "Remaining," to indicate how much of the budgets of each advertiser remaining, and how many displays of each query remain unallocated.

So let's get started with our greedy allocation.

So to start, we go up to our Average Price Per Display table, and we find the combination of advertiser and query which gives us the largest average price per display.

So, in this case, the largest average price per display is five.

And this happens when we display Verizon's ad with Query 3.

So this is the first variable that we're going to change in our greedy allocation.

These are the first, basically the first displays that we're going to use.

So if we scroll down, we add Verizon in Query 3.

And to figure out the budget limit, we go to Verizon's remaining budget which is $160.

And so we take that $160, and we divide it through by the average price per display of $5.

So, we get this number 32, and so what this means is that we can display Verizon's ad with Query 3, 32 times based solely on the remaining budget of Verizon.

Now, of course, we also have to respect how many displays are unallocated of each query, and for query 3, we have 80 displays.

So the query limit at this stage is 80.

So now how many times do we actually display Verizon's ad with Query 3?

Well, the most that we can display Verizon's ad with Query 3 is going to be the smaller of these two numbers.

And so in this case the smaller of the two numbers is 32.

So, we allocate Verizon's ad to Query 3, 32 times.

So now after we've changed the value of this variable, you can see that the budget of Verizon has changed.

So now it is 0.

So we've completely extinguished Verizon's budget, and the remaining unallocated displays for Query 3 has also changed.

So this number used to be 80 and now it is 48.

So, since we've extinguished Verizon's budget, we go back up to our Average Price Per Display table.

And so now we have to select a new combination to use in our allocation.

Since we've used up Verizon's budget, we're going to highlight these cells corresponding to Verizon.

We're going to highlight them in red to indicate that we can't use any of those combinations anymore.

And from the remaining cells in the table, we want to find the combination of Advertiser and Query that gives us the largest average price per display.

So in this case, the next highest is for T-Mobile and Query 3, and in this case the average price per display is two.

So this is the next combination that we'll use in our allocation.

So if we scroll down, we add T-Mobile and Query 3.

So now we have to calculate the budget limit.

T-Mobile's remaining budget is $100.

We take $100, and we divide it through by $2 per display, and we get this value of 50.

And the query limit for T-Mobile and query 3 is the remaining number of displays for Query 3, which is 48.

So we scroll up and we add 48.

And so now, how many times do we actually display T-Mobile's ad with Query 3?

Well, that number is 48, because 48 is the smaller of the two numbers.

So we go ahead and we add that.

And so now T-Mobile's budget has changed.

So it's dropped from 100 to four, and the remaining displays of Query 3, that number has dropped to 0.

So there are no more displays of Query 3 remaining that we can use.

So now we move on to the next stage of our allocation, and we go back to the Average Price Per Display table.

And since we've used up all of the displays of Query 3, we're going to highlight the remaining cells corresponding to Query 3 in red, just to remind us that we can't use them.

And so now from the remaining cells, again, we want to find the highest average price per display combination.

And so from the remaining cells, the highest average price per display occurs when we use T-Mobile's ad with Query 1.

So that's the next variable that we'll use in our allocation.

So we scroll back down and we add T-Mobile with Query 1.

To compute the budget limit, we take T-Mobile's remaining budget which is four, $4, and we divide it through by the average price per display of $1.

So, we get four displays according to the budget.

And the query limit, in this case, is 140, because there are 140 displays of Query 1 that haven't been used towards any ad.

Now how many times do we actually display T-Mobile's with Query 1?

Again, we just take the minimum of these two quantities, which is four.

We go ahead and we enter that into our table.

So now if you look at the budgets, we've completely extinguished T-Mobile's budget and the remaining number of displays of Query 1 that haven't been allocated has dropped to 136.

So now we move on to the next stage of our allocation.

Again, we've eliminated T-Mobile's budget.

So now we highlight those cells in red.

And so now, we want to pick the highest average price per display from the remaining cells.

Now in this case, the only two combinations that remain to us are to display AT&T's ad with Query 1 and to display AT&T's ad with Query 2.

Both of these have the same average price per display, so it doesn't matter which one we really choose.

But for the purpose of this solution, let's just go with AT&T displayed with Query 1.

So the average price per display is 0.5.

So we go down, we add that entry to our side table here.

And to compute the budget limit, we take AT&T's remaining budget of 170, and we divide it through by 0.5.

To get the query limit, we just look at the number of remaining displays of Query 1, which is 136.

So we add that.

And so now, obviously, the smaller of the two quantities is 136.

So we'll display AT&T's ad with Query 1, 136 times.

So now if we look at the state of our allocation, we've basically used up all the displays of Query 1.

And we've used up some part of AT&T's budget, though there's still a lot left.

So now we go back up and we now proceed to make the next allocation in our greedy solution.

[illegible]

We've eliminated AT&T in Query 1.

So now the only combination that remains is AT&T with Query 2.

So we go down.

We add that entry to our side table.

And so now to get the budget limit, we take AT&T's remaining budget of $102, and we divide it through by 0.5, which is the average price per display to get 204.

And the query limit now is 80, because 80 is the number of remaining unassigned displays of Query 2.

And now the smaller of the two numbers is 80.

And so we add 80 to our table.

And so at this point, we've gone through all the entries in our average price per display table, and we've eliminated basically all of them.

And in terms of the allocation, we know that we can't make any more allocations, because if we look at our constraints, basically, there are no more displays of any query that have not been assigned to any advertiser.

So all of these entries here are 0.

And in this case, AT&T's budget is still some positive value.

So we still have $62 of ATamp;T's budget remaining.

But since we don't have any more displays of any query that we can use, we are basically done.

So this is our greedy solution.

And so there are a couple of interesting things to note about this greedy solution.

So the first is that the actual combinations that are used by the greedy solution are different from those that are made by the linear optimization based solution.

So for example, in the optimization solution, if you recall, we only used Query 3 with AT&T.

So we only displayed-- whenever we displayed an ad with Query 3, it was only ATamp;T's ad.

But in this case, we don't actually display AT&T's ad with Query 3 ever.

And in fact, we display only T-Mobile's ad and Verizon's ad with Query 3.

We don't use AT&T. So that's-- so the actual allocation changes and so as a result, the revenue that we get from the greedy solution is different from the revenue that we get from the optimization solution.

If you recall, the revenue from the optimization solution was $428.

Now, this may not seem like a very large difference.

So this is a difference of $60.

But in relative terms, this is actually a rather large amount, 368 relative to 428 is roughly 14 percent.

And so, hopefully this illustrates the fact that an optimization based solution can provide a significant difference in performance relative to a basic common sense solution.

So this concludes our construction and our discussion of the greedy solution.

In the next video, we will go back to our linear optimization model, and we will consider some of the sensitivity analysis that goes along with that model.