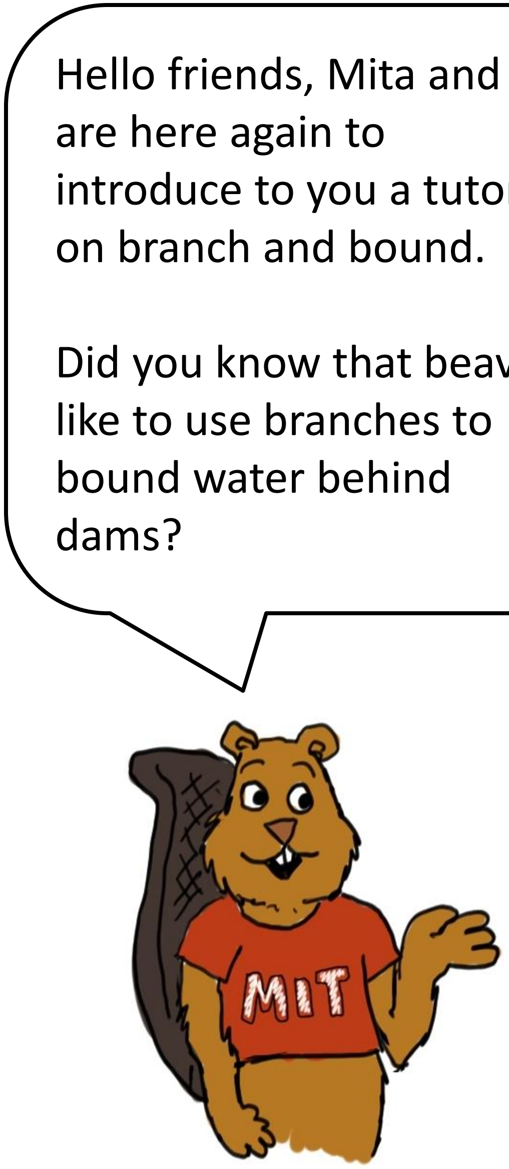
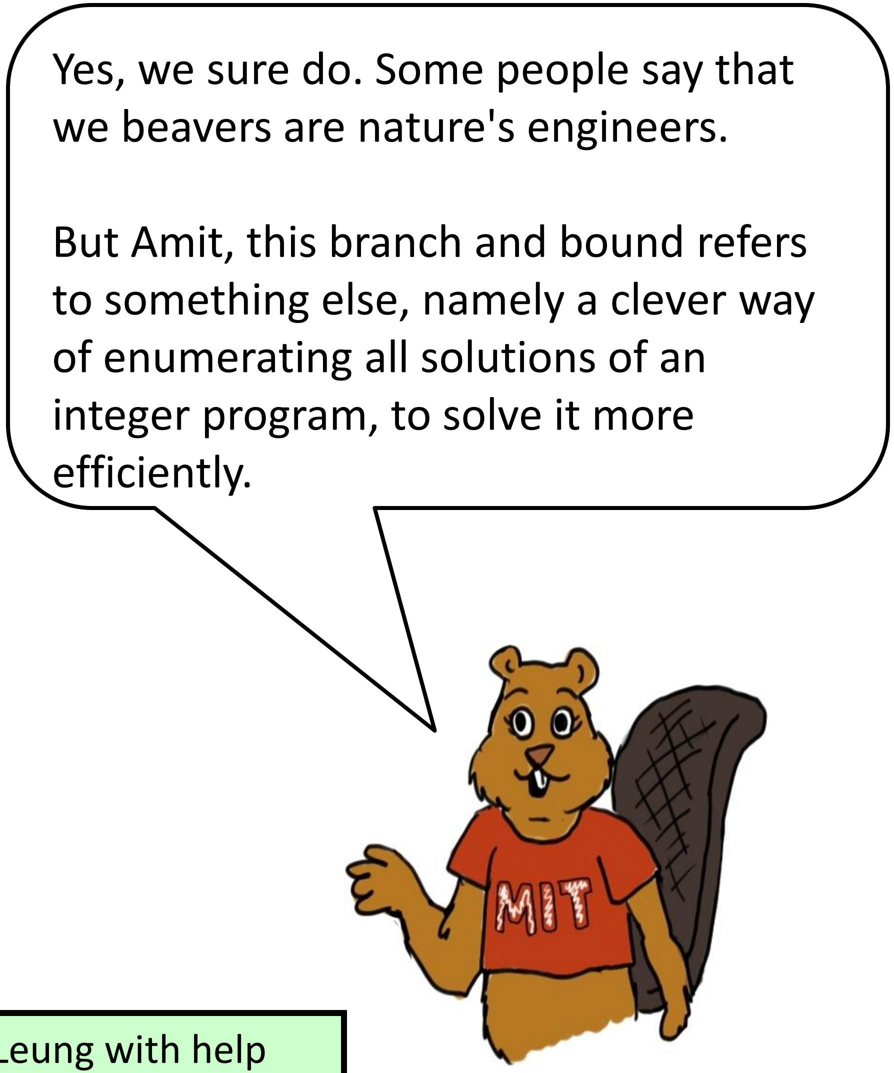
Branch and Bound


Hello friends, Mita and I are here again to introduce to you a tutorial on branch and bound.

Did you know that beavers like to use branches to bound water behind dams?

Amit


Yes, we sure do. Some people say that we beavers are nature's engineers.

But Amit, this branch and bound refers to something else, namely a clever way of enumerating all solutions of an integer program, to solve it more efficiently.

Mita

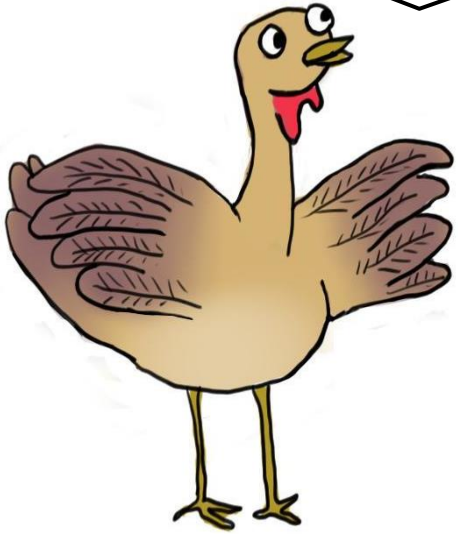
Written by Zach Leung with help from the 15.053/8 team of cartoon characters. April, 2012

$$\begin{array}{ll} \text{maximize} & 15x_1 + 12x_2 + 4x_3 + 2x_4 \\ \text{subject to} & 8x_1 + 5x_2 + 3x_3 + 2x_4 \leq 10 \\ & x_k \text{ binary for } k = 1 \text{ to } 4 \end{array}$$

I still don't really understand how to use the Branch and Bound algorithm. Ella, do you mind going through it once again for me? Suppose I were a contestant on the famous TV game show "*I Heart Trading For Profit*" (IHTFP) that Nooz was on recently (exclusively on the *Fox* TV network). How would I use Branch and Bound in order to solve the problem of what are the best items to pick to maximize my utility?

Of course we can help, *that's what friends are for*.

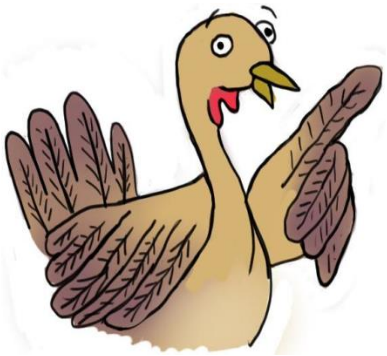
da da da da da, du du du du du du du, Well you came in loving me…

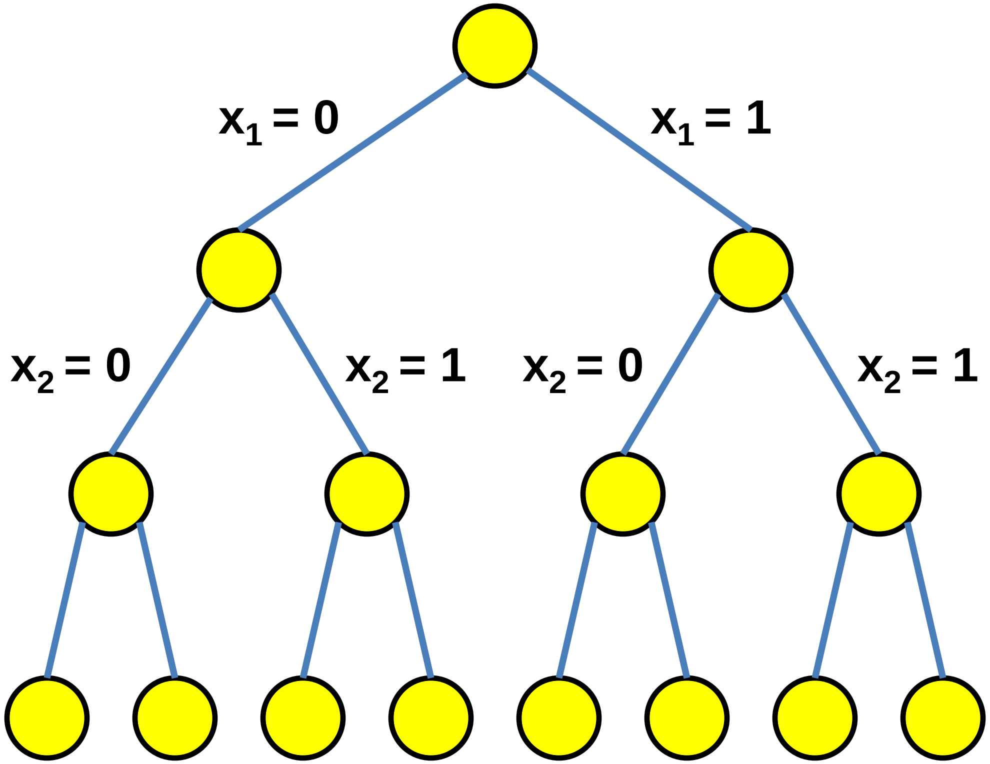


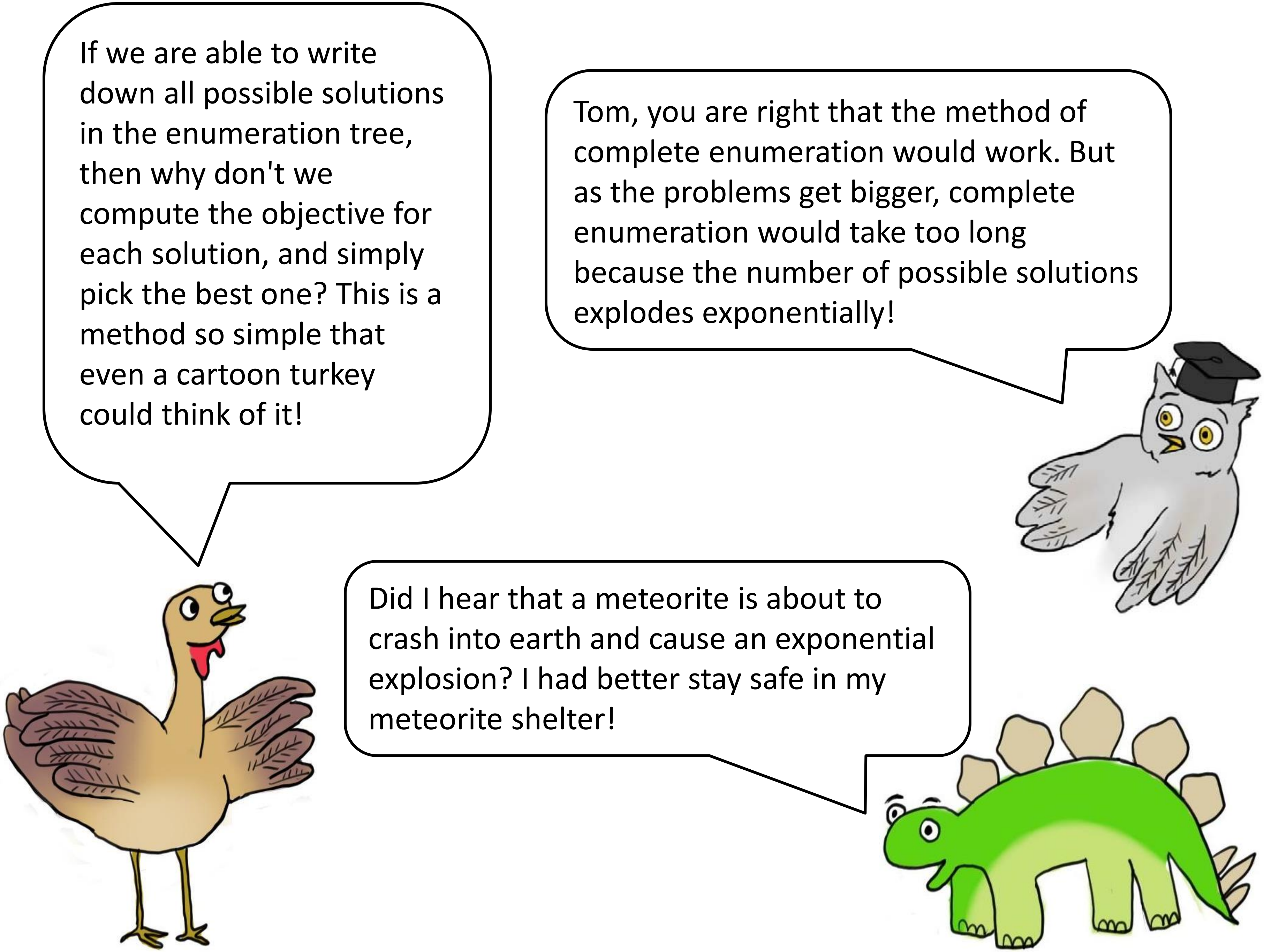
Note: This example differs from the one in the lecture. That is why the trees are not the same.

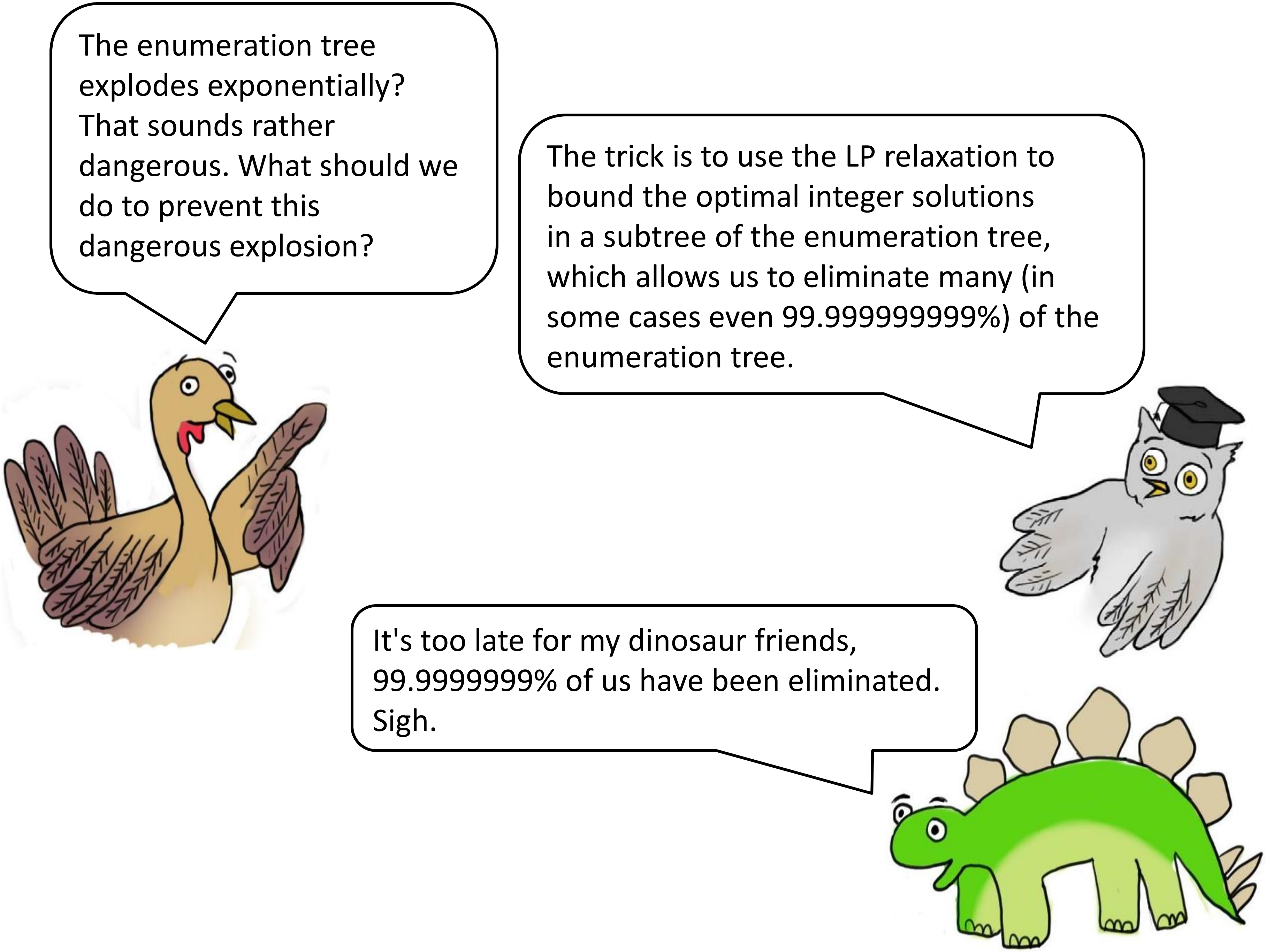
Ella, what are you singing? I don’t recognize the song…

Oh *the young ones*, that was a song before your time… Anyhow, our starting point is the enumeration tree, which is a method to enumerate all possible solutions of an integer program.
At each node, we branch on an integer variable, where on each branch, the integer variable is restricted to take certain values.



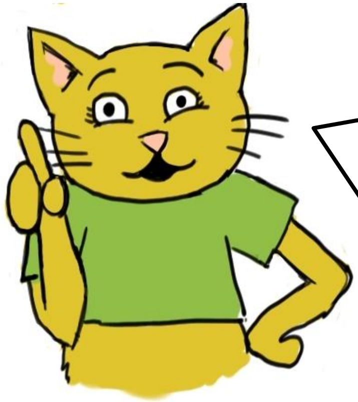



If we are able to write down all possible solutions in the enumeration tree, then why don't we compute the objective for each solution, and simply pick the best one? This is a method so simple that even a cartoon turkey could think of it!
Tom, you are right that the method of complete enumeration would work. But as the problems get bigger, complete enumeration would take too long because the number of possible solutions explodes exponentially!
Did I hear that a meteorite is about to crash into earth and cause an exponential explosion? I had better stay safe in my meteorite shelter!


The enumeration tree explodes exponentially? That sounds rather dangerous. What should we do to prevent this dangerous explosion?
The trick is to use the LP relaxation to bound the optimal integer solutions in a subtree of the enumeration tree, which allows us to eliminate many (in some cases even 99.999999999%) of the enumeration tree.
It's too late for my dinosaur friends, 99.9999999% of us have been eliminated. Sigh.

Essential Branch and Bound

I will summarize in one slide the branch and bound algorithm!
To start off, obtain somehow (e.g. by extortion, creativity, or magic) a feasible solution x^*. At each iteration of the algorithm, we will refer to x^* as the *incumbent solution* and its objective value z^* as the *incumbent objective*. Here, incumbent means “best so far.”
Next, mark the root node as active.

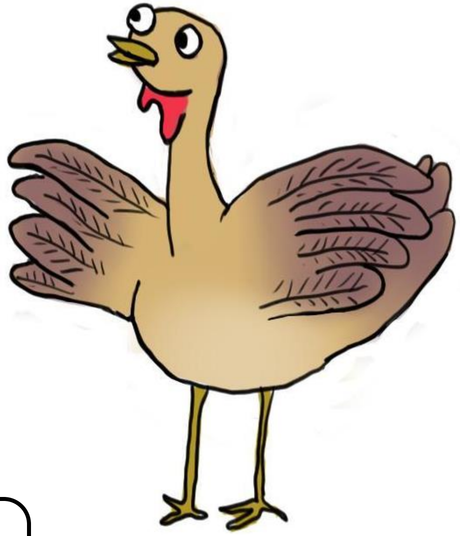


```
While there remain active nodes
    Select an active node $j$ and mark it as inactive
    Let $x(j)$ and $z_{LP}(j)$ denote the optimal solution and
    objective of the LP relaxation of Problem($j$).
    Case 1: If $z^* \geq z_{LP}(j)$ then
        Prune node $j$
    Case 2: If $z^* < z_{LP}(j)$ and $x(j)$ is feasible for IP then
        Replace the incumbent by $x(j)$
        Prune node $j$
    Case 3: If $z^* < z_{LP}(j)$ and $x(j)$ is not feasible for IP then
        Mark the direct descendants of node $j$ as active
End While
```

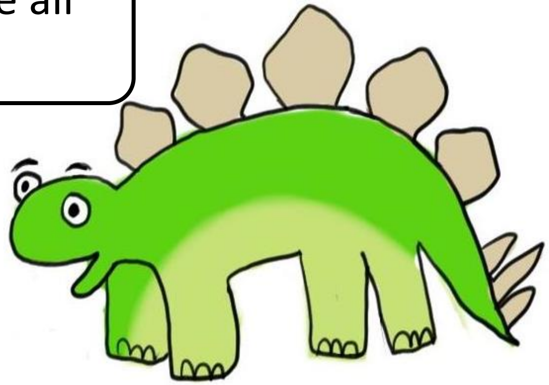
Let’s get started! Big money, big money, *spin*

In your case, Tom, we shall start with the incumbent solution $x^* = (0,0,0,0)$ and incumbent objective $z^* = 0$.

That doesn’t seem like a very good solution. I would not be happy with this solution at all, but go on. I’m sure things will only get better from here.



Unfortunately, my dinosaur friends are all getting exactly zero.

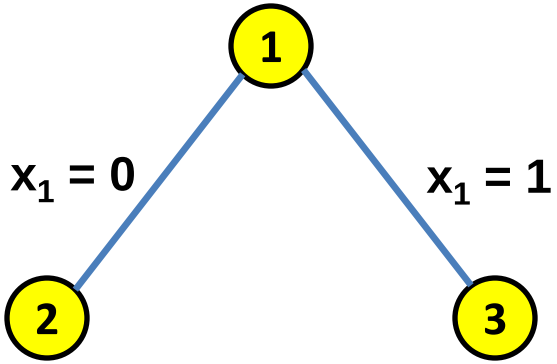


Incumbent

$$x^* = (0, 0, 0, 0)$$

$$z^* = 0$$

Let’s start at the very beginning, a very good place to start. Our starting point is the single node containing all feasible solutions.



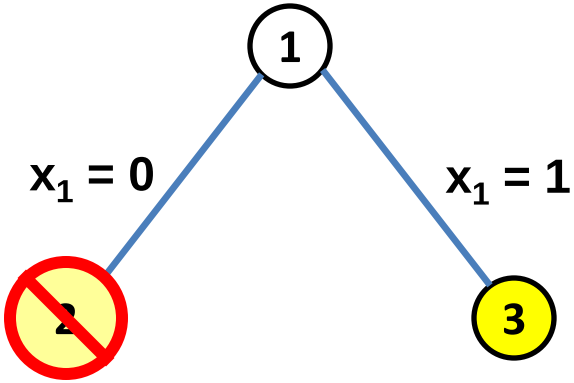
Incumbent

$$x^* = (0, 0, 0, 0)$$
$$z^* = 0$$

$$x_{LP}(1) = (5/8, 1\ , 0, 0)$$
$$z_{LP}(1) = 21\ 3/8$$

We solve the LP relaxation of problem 1.

Since we are in case 3, we branch on the node and mark its children as active.



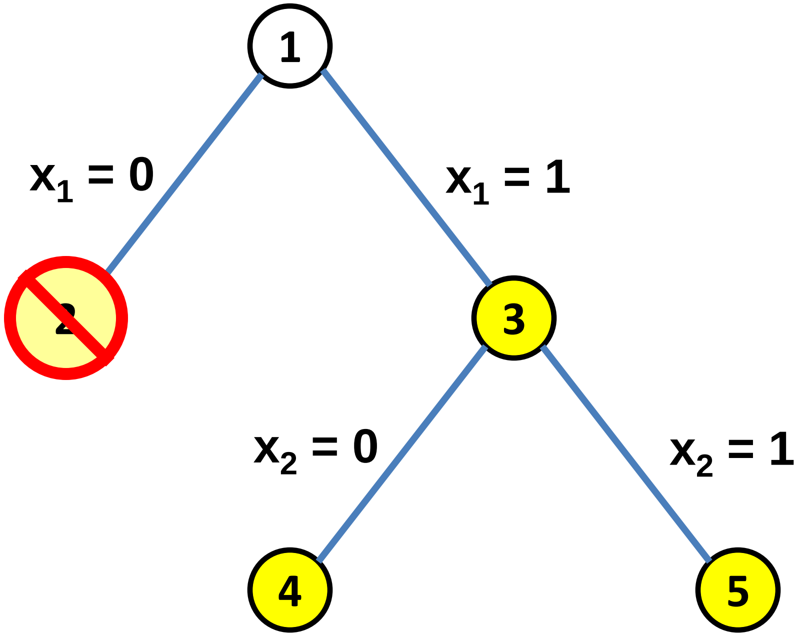
Incumbent

$$x^* = (0, 0, 0, 0)$$
$$z^* = 0$$

$$x_{LP}(2) = (0, 1, 1, 1)$$
$$z_{LP}(2) = 18$$

We solve the LP relaxation of problem 2.

Since we are in case 2, we replace the incumbent with this better solution, and prune node 2.



Incumbent

$$x^* = (0, 1, 1, 1)$$

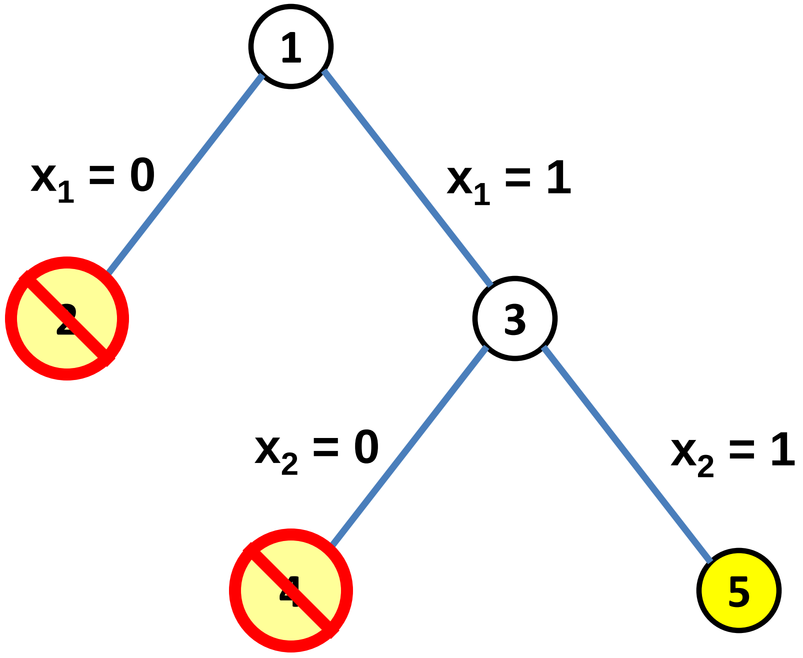
$$z^* = 18$$

$$x_{LP}(3) = (5/8, 1, 0, 0)$$

$$z_{LP}(3) = 21\ 3/8$$

We solve the LP relaxation of problem 3.

Since we are in case 3, we branch on the node and mark its children as active.



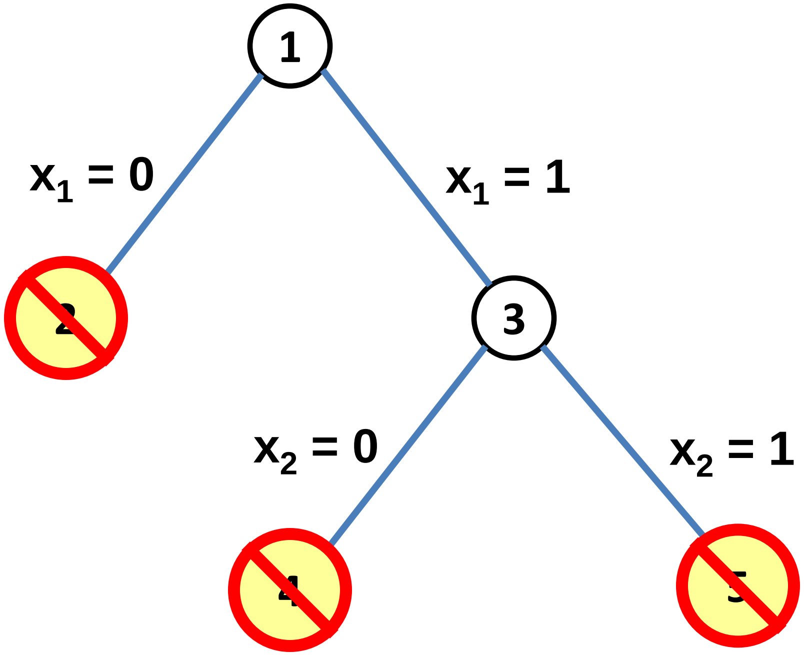
Incumbent

$$x^* = (0, 1, 1, 1)$$
$$z^* = 18$$

$$x_{LP}(4) = (1, 0, 2/3, 0)$$
$$z_{LP}(4) = 17\ 2/3$$

We solve the LP relaxation of problem 4.

Since we are in case 1, we prune node 4.



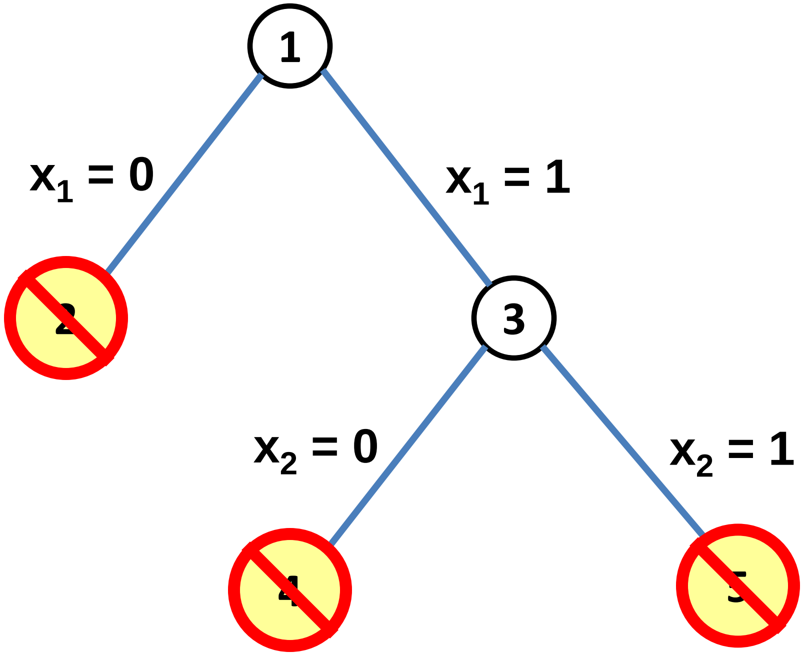
Incumbent

$$x^* = (0, 1, 1, 1)$$
$$z^* = 18$$

$$x_{LP}(5) = \text{infeasible}$$
$$z_{LP}(5) = \text{infeasible}$$

We solve the LP relaxation of problem 5.

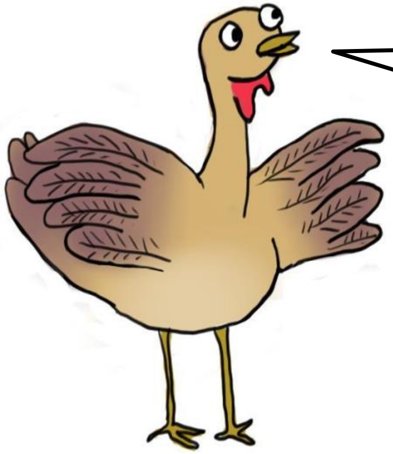
Since we are in case 1, we prune node 5.



Incumbent

$$x^* = (0, 1, 1, 1)$$

$$z^* = 18$$



And we're done! Yahoo! Google! Bing! Thank you Ella.

You're most welcome. That's what friends are for… do do do do do

Although branch and bound looks difficult at first, just like building dams, it gets easier with practice. To share a motivating example from my own experience: When I was younger, I thought building dams was dam hard, but after working dam hard at it, I now find it to be dam easy!

That’s all for branch and bound. Bye!

Amit

Mita