Theory of Travel Computation
by Rick Drushal
So one day Bob decided that he was going to take a vacation....

I’m tired of working. I need a break!!!!

Maybe I will take a vacation!

That was the easy part of Bob’s decision.
Now Bob had to decide where he wanted to go.

So Bob thought... and thought some more... and thought some more...

(Bob is kind of slow, so you see where this is going)
Eventually, Bob decided that he wanted to visit London, Cairo, Tokyo, San Francisco, and Anchorage.

But Bob is also a cheap skate, and he wanted to complete his trip as efficiently as possible.
So, Bob got out a map and put pins in everywhere that he wanted to go.

But just looking at the map and guessing the shortest and cheapest routes wasn’t good enough for Bob.
So, he found out how much it would cost to get from each city to every other city.

But, Bob quickly decided that drawing lines and just looking at the map would not work. He needed something more technical to assure that he got the very best deal.
So Bob went and dug through his attic until he found his old friend: *Sipser’s Theory of Computation* and *Introduction to Algorithms*
From his Algorithms book, Bob read how to create a weighted edge graph. He realized that it would probably look something like this:

Then, Bob assumed that he would just look at the edges and pick the cheapest ones to get from his home in New York to each place and back to New York again.

(But with the weights on each of the edges which would make it too complicated to display well)
Bob did this a few times and got a lot of different answers.... But he wasn’t sure if any of them were the very best.

I must have the very best.
I will settle for nothing less!

So, he kept going... and going... and going.... Long enough that the Energizer™ Bunny would have gotten tired.
Finally, Bob got tired of doing all of this by hand and thought to himself:

Self, there has to be a better way to do this. In college we used matrices to represent graphs so computers could work on them.
So, Bob sat down at his computer and put all of the entries from his graph into a matrix and started writing an algorithm to find the cheapest way to take his trip.

Hmmmm... this is a difficult algorithm to create... But I know I can do it!!
Finally, Bob came up with an algorithm that he was sure would solve his problem. It was foolproof.

So, Bob compiled his program and got ready to run it.
(The amazing thing was that it actually compiled)
Then, Bob went to the terminal and decided to run his CheapestTrip Program.

Bob ran his program....

And waited...
And waited...
And waited...
And waited...
Unfortunately for Bob... The only response he ever got to his program was the following message:

"Bob... I know you programmed me, but it is impossible to find an efficient algorithm to find the absolutely cheapest trip. If you had been in class that day instead of sleeping in you would have remembered about problems which can only be solved in Non-Deterministic Polynomial Time. Please go look at this section in your book, and then give up on this algorithm."

Wow... that's a smart computer...
Dejected that his Algorithm didn’t work, Bob opened Sipser’s book for the first time.
(Since he never really used it while he was in college)

He read about NP problems. Bob was sad.

It turned out that his Computer was right. He found out that if a problem is NP-Complete, it can’t be calculated in Polynomial time, based on the size of the input. This means that even Bob’s small Graph is very hard to analyze.
Bob got more and more dejected as he read this. He kept thinking that there would be no way he could ever guarantee that he was going to be going on the cheapest trip.

Unfortunately for Bob, he was unwilling to settle for something that \textit{might} be the cheapest trip.

\begin{quote}
Wow.... I wish I was smart enough to create a Polynomial time algorithm for this problem.
\end{quote}

So, since he could not devise a polynomial time algorithm for this problem...

By the way, this would have make him very famous (for proving that P = NP)....
Bob decided that it wasn’t worth the risk of spending extra money to go on the trip. He decided to heat up a Dijournos’s pizza and just stay at home.

It’s too bad that Bob couldn’t solve the algorithm, but then, people have been trying to write polynomial time algorithms for problems like these for a long time.
And, like we said in the beginning, Bob wasn’t very bright to begin with.

I mean look at that vacant stare....

If only Bob had been able to live with an approximation....
The moral of the story is that sometimes it is better to be close than to be exact...

And not just in Horseshoes and Hand Grenades

But only if it is an NP problem. For instance, just finding a Hamiltonian Path.

(let alone finding an optimal one)
The End