

Day 12

Fixing Santa's Navigator

The Story

Santa was livid. His face was redder than Rudolph's nose after a Red Bull binge. The elves had tried the sleigh's navigation system again, running the `find_best_location` function. Instead of landing them in a cozy, gift-ready location, the sleigh had whisked them right back to the North Pole.

The Story

"I told you to land in the best spot, not bring me back to this frosty hellhole!" Santa bellowed. His voice echoed across the frozen tundra, sending a nearby snowman into early retirement.

The Story

Bernard cautiously stepped forward.
"Erm... Sir, the function seems to be, uh, calculating the densest snow coverage but doesn't account for it's distance from us." Santa's eyes narrowed. "I said best place to land, not snowiest! Who do I look like, a yeti?"

The Story

"Show me the code," Santa growled,
extending a gloved hand that looked ready
to smash a keyboard.

Bernard shakily handed him the tablet.

"Here it is, boss."

The Story

Santa squinted at the screen, his rage escalating as he scanned the lines. "FOR HOLLY JOLLY'S SAKE—are you not using `references` again?! A consuming iterator? Do you want to crash the sleigh into a snowbank?!"

The Story

The elves collectively winced, knowing that Santa's rants about proper memory management could last for hours.

Santa shoved the tablet back at Bernard. "Fix it, or I'm replacing the sleigh team with drones. And, use references for the love of all that's holly and jolly, Bernard! I won't tolerate another memory leak on Christmas Eve!"

Your Mission

The function `find_best_location` is not behaving as expected. But the code might still be useful, so the elves don't want to scrap it entirely, instead they renamed it to `find_most_dense_location`.

Here is what you need to do for today:

- Update the `functions` signature to accept a `&[Location]` instead of a `Vec<Location>` and return a reference to a `Location` instead of an owned value.

Your Mission

- Write the code for the other function `find_nearest_location` that will return the nearest location to the current location which is `(x = 0, y = 0)`. Ignore the `z` coordinate.
- There should be a minimum of density of `1000.0` to be considered a good location. If there weren't any, return an `Error`.