Daily Coding Problem #248

Problem

This problem was asked by Nvidia.

Find the maximum of two numbers without using any if-else statements, branching, or direct comparisons.

Solution

There are a few different ways to solve this.

First let's try a mathematical approach. Suppose we have two numbers `a` and `b`. We can separate our problem into two cases: either `a < b` or `b < a`. (If they are equal, either number will be a valid solution).

Suppose the first case is true. Then we can call the positive difference `b - a` by the variable `d`. It follows that `a + b + d` will be twice the maximum. On the other hand, suppose the second case is true. Then the difference `b - a` would have the same magnitude but would be negative. For example, `10 - 8 = 2`, while `8 - 10 = -2`. In this case, we would need to make `d` positive in order for `a + b + d` to equal twice the maximum.

To make both cases work, then, we can take the absolute value of the difference, add it to the sum of both numbers, and divide by two.

```
def max(a, b):
    return 0.5 * (abs(b - a) + (a + b))
```

Another solution can be found using bitwise operations.

Consider the following expression: `a - k * (a - b)`. If `k = 0`, this returns `a`, whereas if `k = 1`, this returns `b`. So if we had some way to assign `k` to one of these values based on which term was greater, we would have a solution. Fortunately, we can do this by shifting `(a - b)` all the way to the right and taking the most significant bit.

Any positive difference will be shifted down to zero. Meanwhile, a negative difference will result in `1`, because negative numbers are typically stored using two's complement representation.

Here is how we could implement this:

```
def max(a, b):
    k = (a - b) >> 31 & 1
    return a - k * (a - b)
```

Privacy Policy

Terms of Service

Press