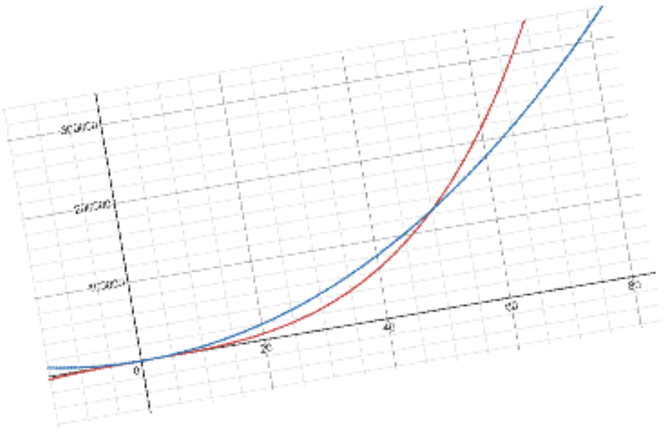
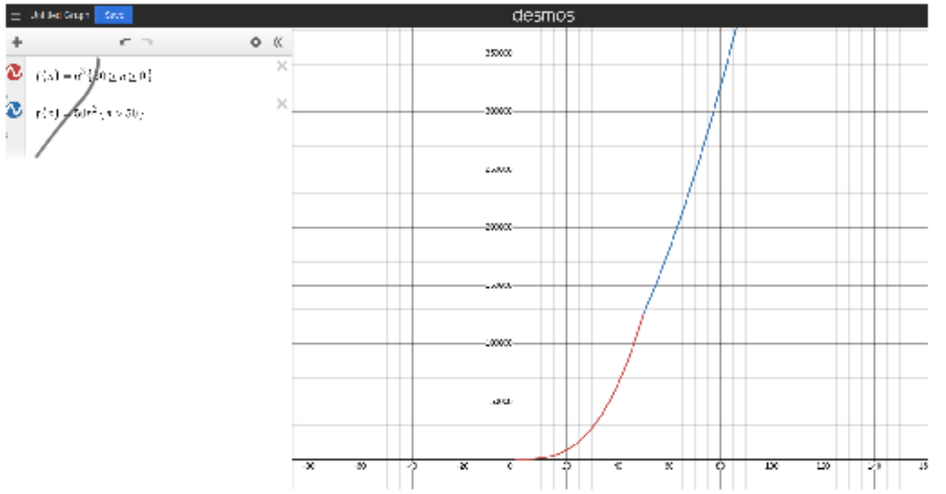
Peace and d.aki are arguing over the tightest upper-bound for the following function. Peace argues that it's $\mathcal{O}(n^2)$ while d.aki argues that it's $\mathcal{O}(n^3)$. Given the following function, who is correct? Justify your answer.

```
def mystery(n: int):
  L = []
  for i in range(n):
      for j in range(min(50, n)):
          L.insert(0, j)
```

Peace is correct!!! frfr ong

Let n_0 = 50 and c = 50 and g(n) = n^2. Then for all n > n_0, we have f(n) <= c * g(n), so O(g(n)) is O(n^2).

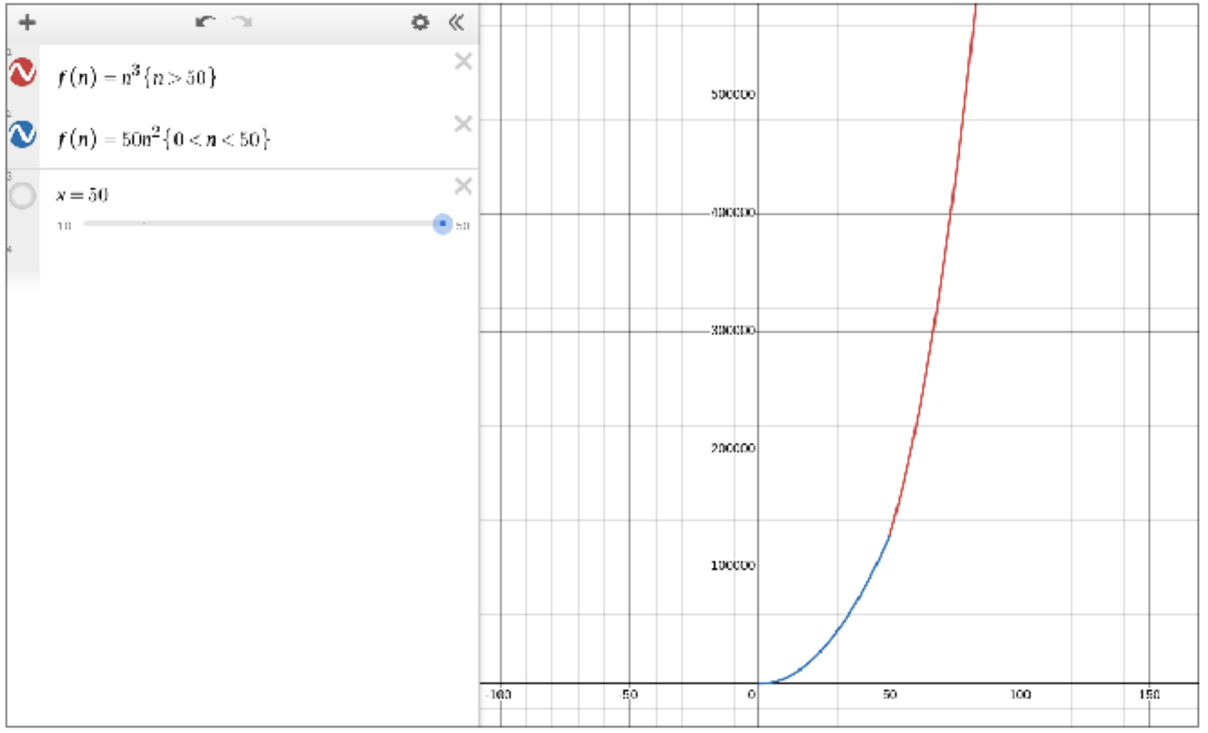




What if we changed the `min` to `max`? Abdulkader thinks it's still $\mathcal{O}(n^2)$, but drew thinks it's $\mathcal{O}(n^3)$. Who is correct. Justify your answer.

```
def mystery2(n: int):
  L = []
  for i in range(n):
      for j in range(max(50, n)):
          L.insert(0, j)
```

O(n^2) fr fr fr fr



```
def foo(x: int):
  n = x
  for i in range(50):
    if n <= 50:
      for j in range(n):
        x += 1
    if n <= 100:
      for j in range(n):
        if n <= 75:
          for k in range(n):
            x += 1
```