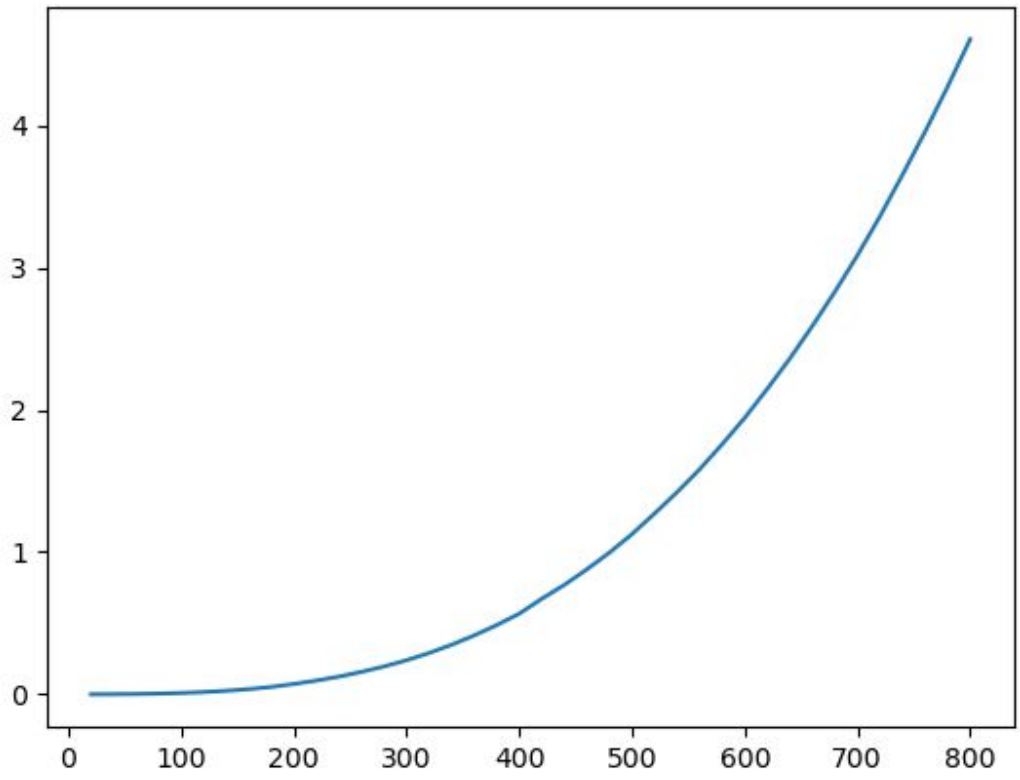
Benjamin Weisman
Noah Solomon

1a) What does the program/algorithm do?
The algorithm takes in a list and permutes it by checking to see if each index in the array is smaller than the next. It repeats this until the list is arranged from largest to smallest.

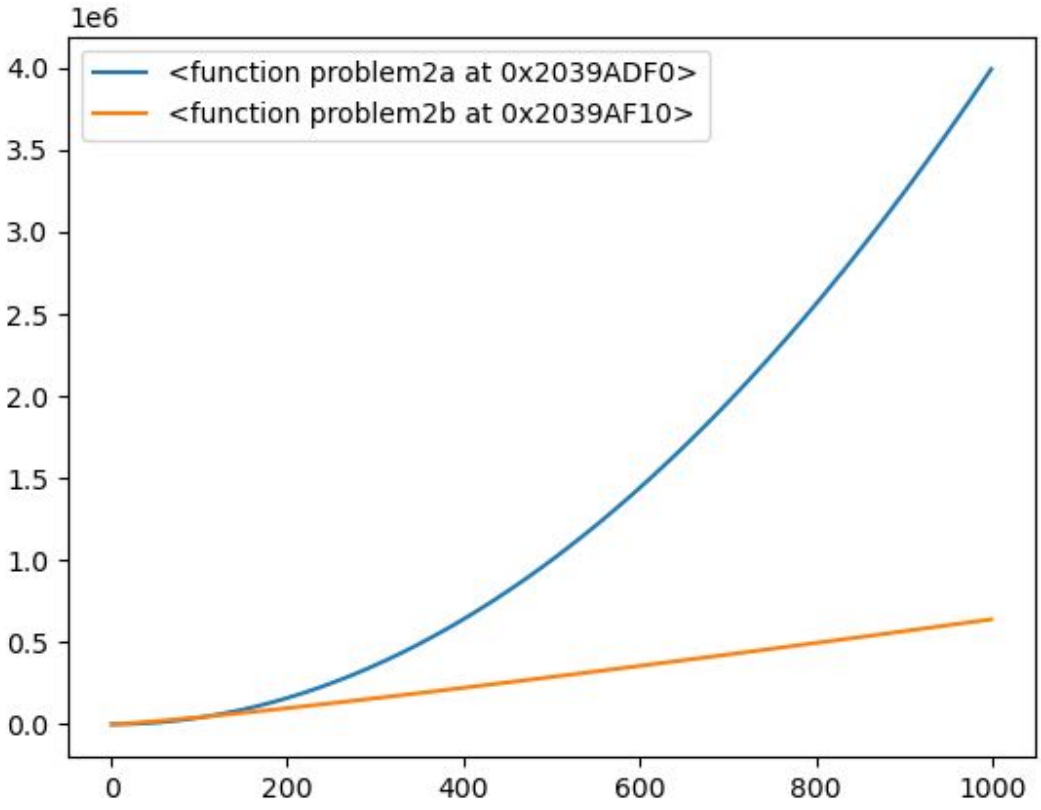
1b) How does the runtime of the algorithm scale on inputs of increasing size?

O(n^2)

1c)



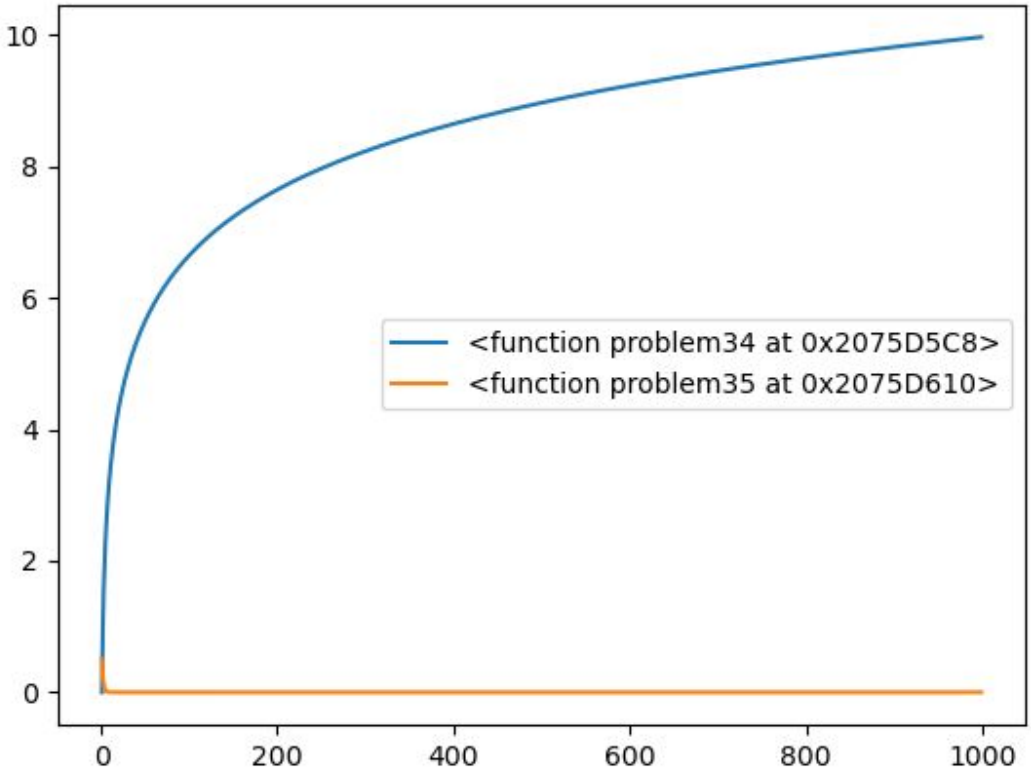
2a)

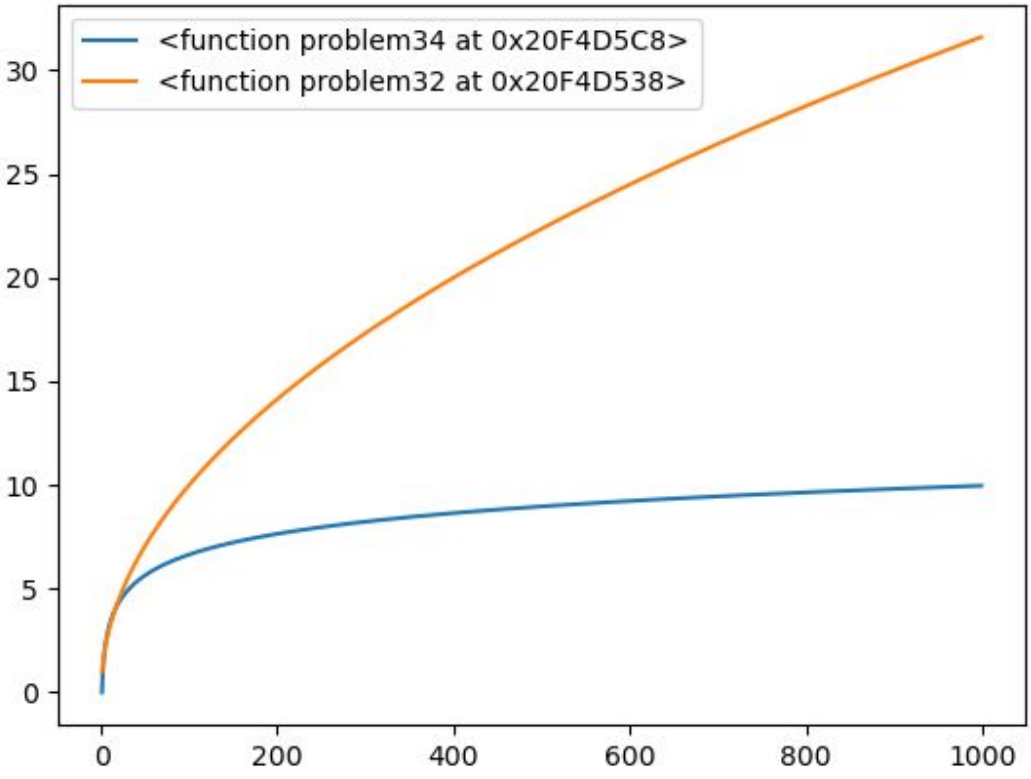


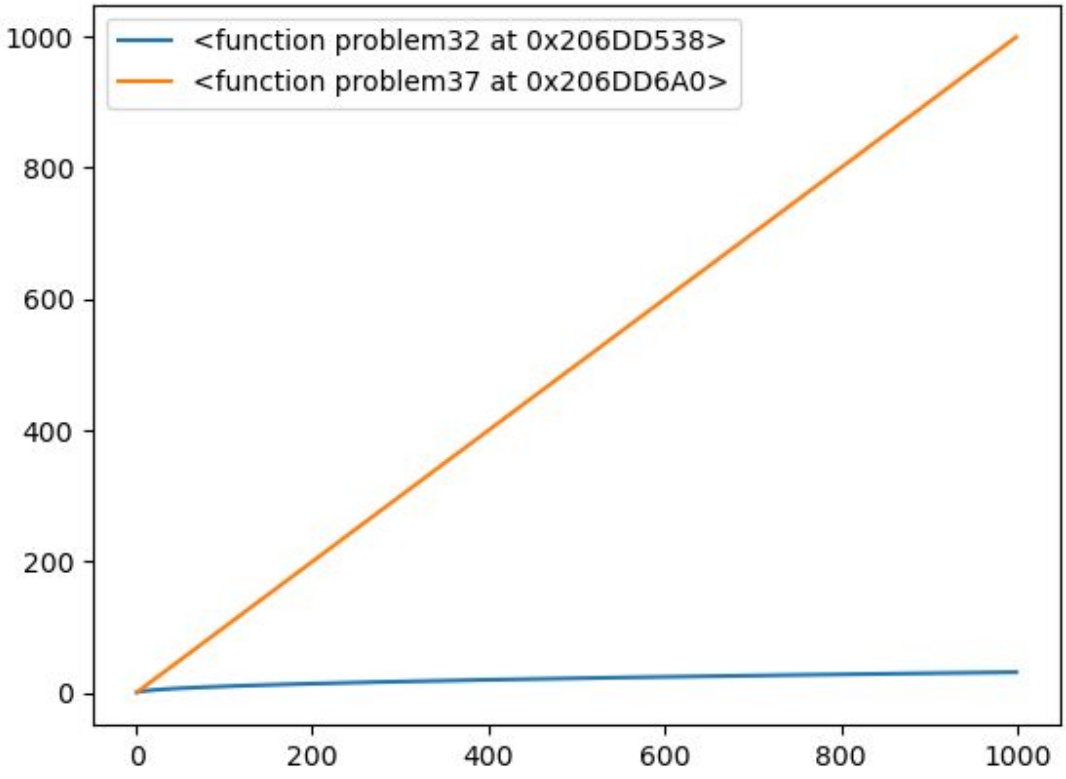
2b) intersects on 107.14465773

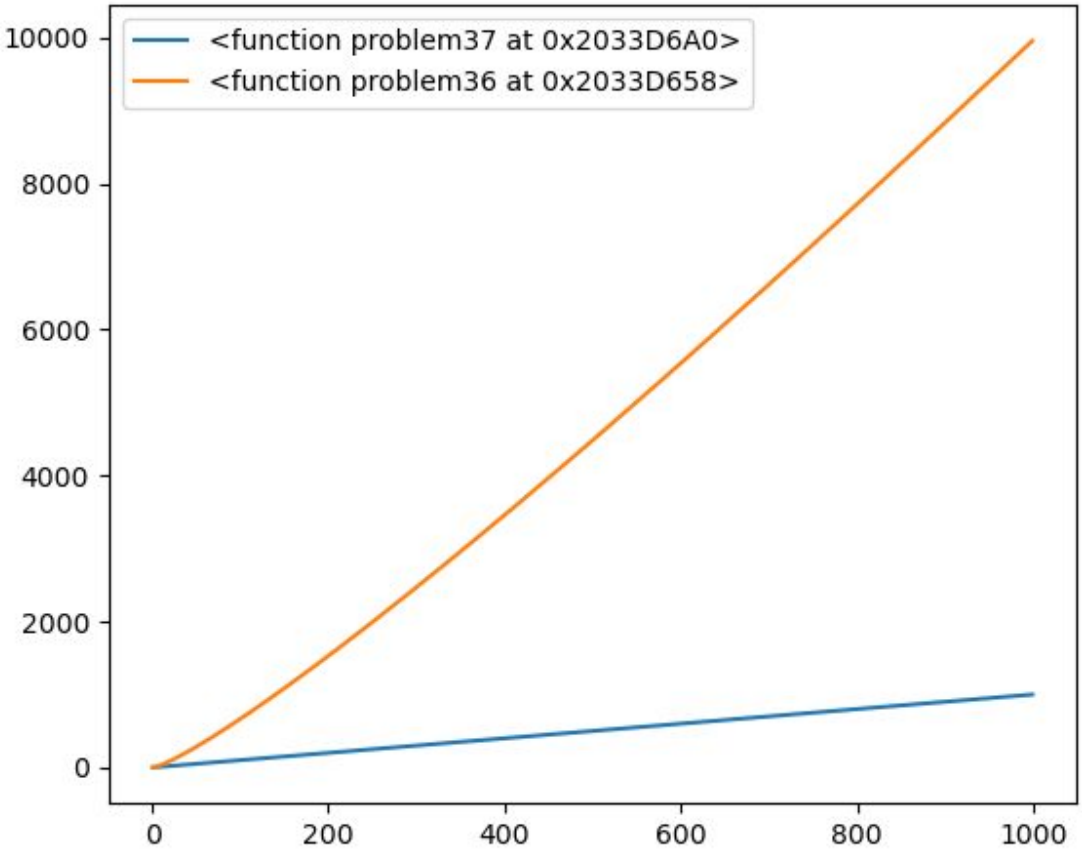
2c) $R_2(n) \in o(R_1(n)) \Rightarrow \forall \varepsilon > 0 \, \exists N \in \mathbb{N}$ s.t. $abs(R_2(n)) \leq \varepsilon R_1(n) \, \forall n \geq N$

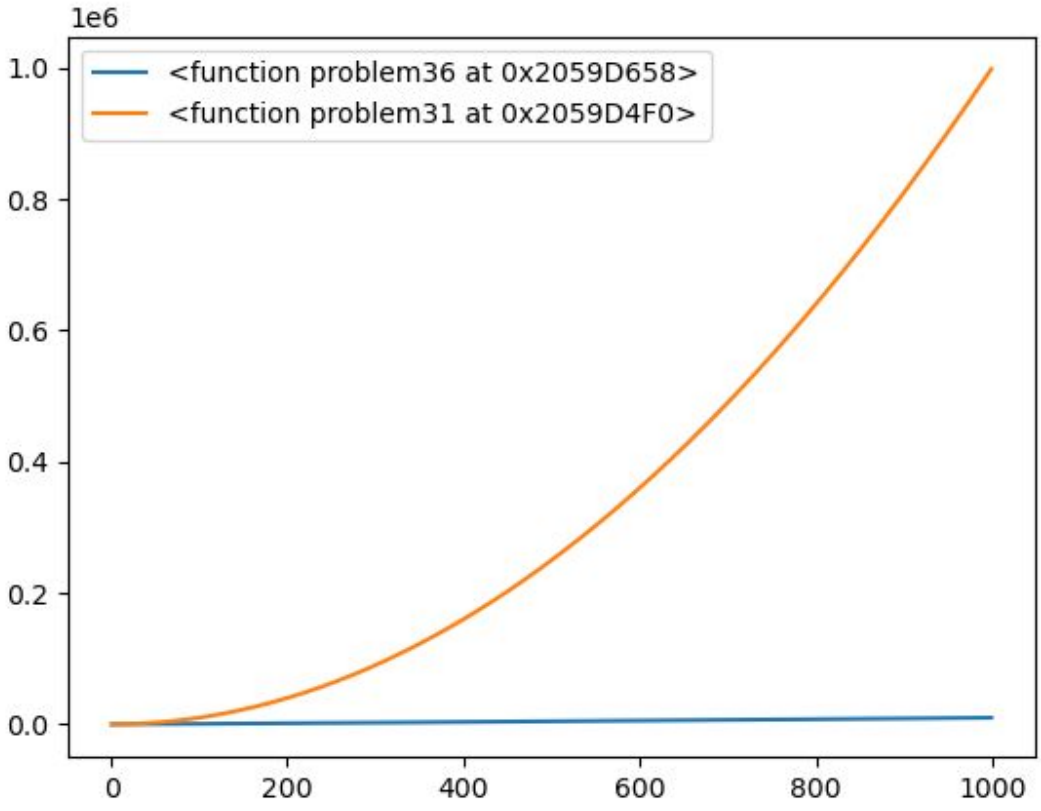
3b) Each function is little o of the function before it.

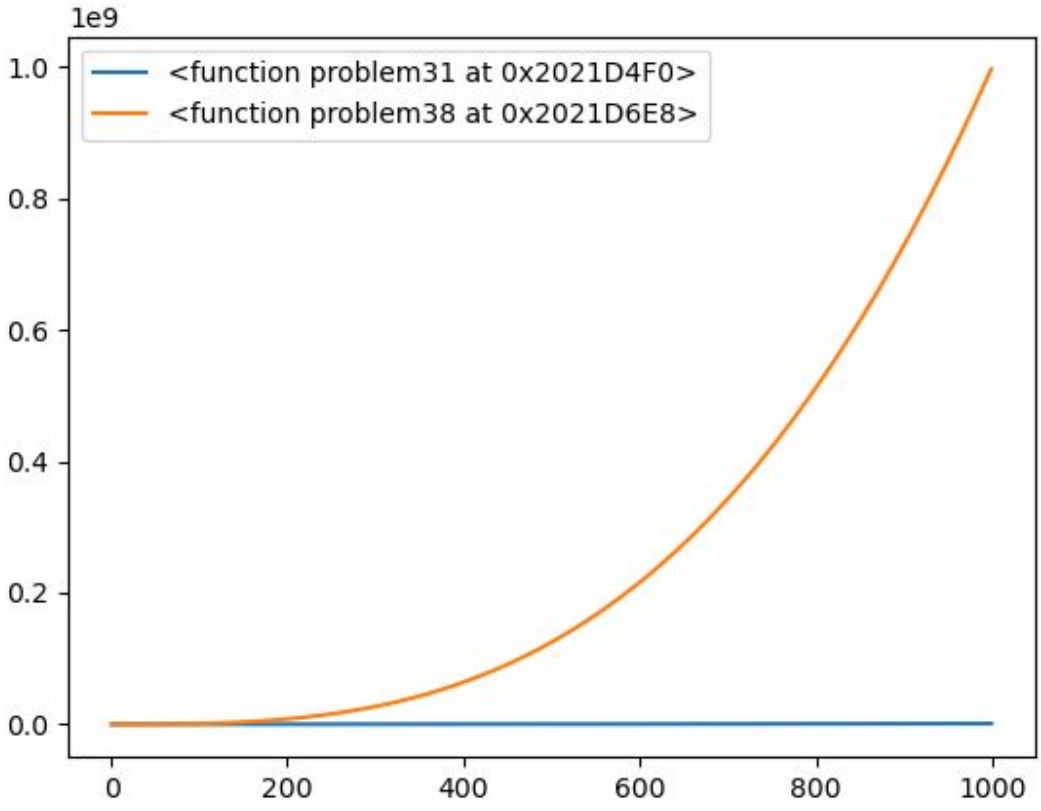


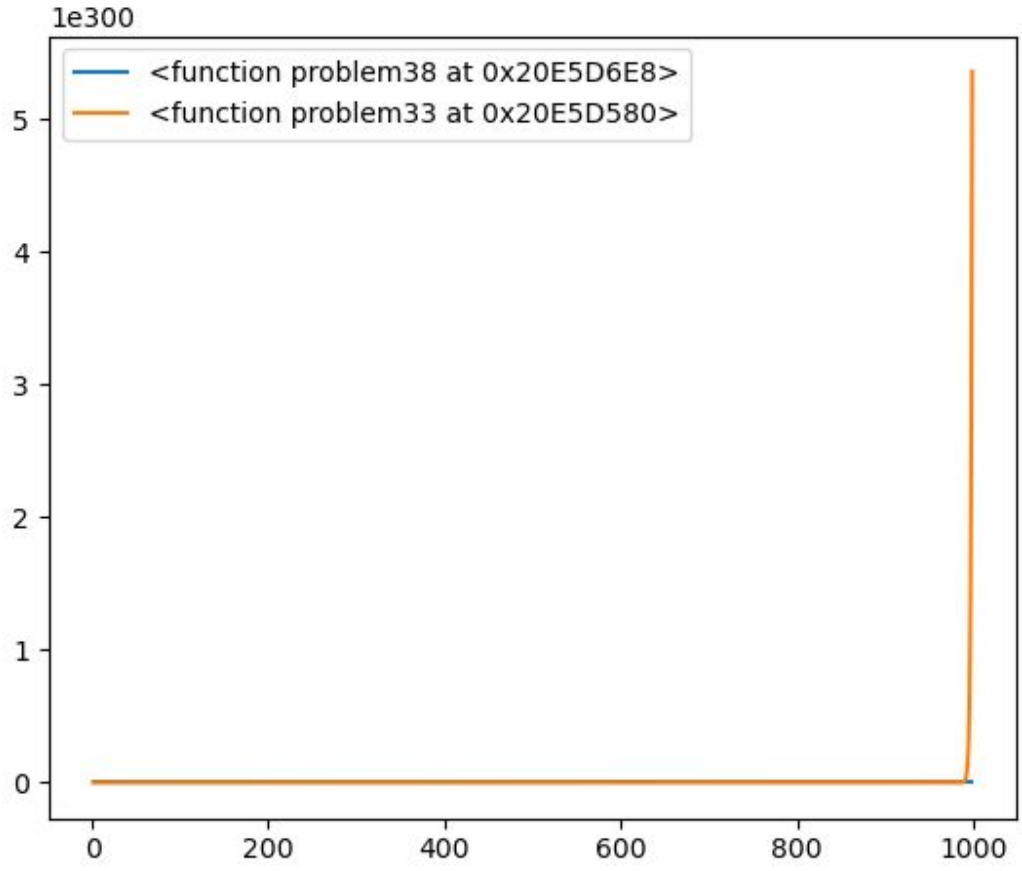

<function problem34 at 0x20F4D5C8>
<function problem32 at 0x20F4D538>
30
25
20
15
10
5
0
0
200
400
600
800
1000


<function problem32 at 0x206DD538>
<function problem37 at 0x206DD6A0>
1000
800
600
400
200
0
0
200
400
600
800
1000


<function problem37 at 0x2033D6A0>
<function problem36 at 0x2033D658>
10000
8000
6000
4000
2000
0
0
200
400
600
800
1000


1e6
1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000
<function problem36 at 0x2059D658>
<function problem31 at 0x2059D4F0>


1e9
1.0
0.8
0.6
0.4
0.2
0.0
0
200
400
600
800
1000
<function problem31 at 0x2021D4F0>
<function problem38 at 0x2021D6E8>


1e300
<function problem38 at 0x20E5D6E8>
<function problem33 at 0x20E5D580>
5
4
3
2
1
0
0
200
400
600
800
1000