Lower Bound Proofs:

1. Equals:

```
growth rate = [10, 10, 10, 10, 5, 5, 5, 5, 5, 5, 5, 5]
queue = [(0, 1), (2, 3), (4, 5), (6, 7), (0, 1), (2, 3), (8, 9), (10, 11)]
```

Highest Lower bound: 35

Cutting one bamboo with a growth rate of 10 and another bamboo with a growth rate of 5 every day is how we alternate between them in this situation. We would have chopped all bamboo with a growth rate of 10 twice and those with a growth rate of 5 once at the conclusion of the periodic timetable. With a growth rate of 5, the bamboo would have gone 7 days without being chopped, giving it a height of 35. On the last 3 days, we get our highest lower bound. Each of the eight such bamboos get chopped once over the full timetable. However, due to our garden's alternating schedule, we observe its height of 40 on six of those days, making this height impossible to achieve.

2. Inequality:

```
growth rate = [98, 98, 1, 1, 1, 1]
queue = [(0, 1), (2, 3), (0, 1), (4, 5)]
```

Highest Lower bound: 98

At day 1: Suppose we cut the trees, with indices 0 and 1, we get 0, 0, and 1,1,1,1 as a growth rate.

So, at the end of that day the growth rate would be 98,98,2,2,2,2.

At day2: Suppose we cut the trees, with indices 2 and 3, we get 98,98,0,0,2,2 as a growth rate.

So, at the end of that day the growth rate would be 196, 196, and 2,2,4,4.

As we need to cut every tree once, it is not possible to cut indices 0 and 1 together every time , so one of them is going to exceed 98 and the maximum height is going to be 196, so that is why 98 is the largest highest that we can't achieve.

3. Split:

```
growth_rate = [100, 32, 16, 8, 4, 2, 1, 1]
queue = [(0, 1), (0, 2), (0, 3), (0, 1), (0, 2), (0, 4), (0, 1), (0, 5), (0, 2), (0, 1), (0, 6), (0, 7)]
```

Highest Lower bound: 64

Each day we can cut max 2 bamboo tress and we can't repeat it same bamboo every single day because we need to achieve minimal optimal solution which we can achieve by cutting each bamboo in that plot, with possible total minimal height.

As we need to cut every tree once, it is not possible to cut indices 0,1, together every time , so one of them is going to exceed 64 and the maximum height is going to be 100, so that is why 64 is the largest highest that we can't achieve.

4. Power:

The input for equals is as below:

Input:

```
growth_rate = [96, 54, 54, 48, 24, 18, 18, 12, 6, 6, 6, 3, 3, 2, 2, 2]
queue=[(0, 1),(2, 3), (4, 5),(0,15),(1,14),(0,2), (3,13),(1,12),(0,11),(2,6),(3,0),(1,4),(7,10),(0,2),(1,8),(3, 9)]
```

Highest Lower bound: 96

Here, the logic behind is to cut a bamboo of growth rate 96 each day and a bamboo of growth rate 54,48 every 3 days for balance. We will never achieve a minimum height of 96 since to do that we would have to cut both bamboos of growth rate 96 every day and in the meantime our bamboos of growth rate would grow indefinitely, surpassing the height of 96. Therefore, we need to also cut the smaller growth rate bamboos periodically. Since we can only cut two bamboos each day, that would mean that one bamboo of growth rate 96 would be left uncut at some point, hence max height 196 (considering initial overnight growth).

5. Fibonacci

```
growth_rate = [55, 34, 21, 13, 8, 5, 3, 2, 1, 1]
queue= [(8, 9), (0, 1), (2, 3), (0, 1), (4, 5), (0, 1), (2, 3), (0, 1), (6, 7), (0, 1), (2, 3), (0, 1),]
```

Highest Lower bound: 68

Each day we can cut max 2 bamboo tress and we can't repeat it same bamboo every single day because we need to achieve minimal optimal solution which we can achieve by cutting each bamboo in that plot, with possible total minimal height.

As we need to cut every tree once, it is not possible to cut indices 1 every time ,so one of them is going to exceed 68 and the maximum height is going to be 110, so that is why 68 is the largest highest that we can't achieve.