Submitted To

Prof. Sebastian Wild

Submitted By

Name: Balkrishna Bhatt
Student Id: 201673048

UNIVERSITY OF
LIVERPOOL

COMP 526 Efficient Algorithms

Programming Puzzle -1 Report

Submitted to:

Prof. Sebastian Wild

Submitted By:

Balkrishna Bhatt
(201673048)

Submitted To

Prof. Sebastian Wild

Submitted By

Name: Balkrishna Bhatt
Student Id: 201673048

Collaborated With

S NO.	STUDENT ID	NAME	COLLABORATION ROLE
1	201648694	Anjaly Krishnan	Worked on the best possible queue for all the required plots
2	201673048	Balkrishna Bhatt	Worked on the lower-bounds proofs and highest lower-bounds for the height that is impossible to achieve
3	201665415	Hitisha Gotecha	Worked on the best possible queue for all the required plots
4	201613334	Prajakta Chindhe	Worked on the lower-bounds proofs and highest lower-bounds for the height that is impossible to achieve
5	201664291	Vanshika Saxena	Worked on the best possible queue for all the required plots

Submitted To

Prof. Sebastian Wild

Submitted By

Name: Balkrishna Bhatt
Student Id: 201673048

OBJECTIVE

To find an infinite schedule of cuts that keeps the maximum height as low as possible.

OUTCOMES

For this problem, I have used the **Greedy Search Approach & Priority Scheduling** **(aren't we all using greedy algo in our day-to-day life to find the best things for us 😂 (just kidding))** to find the lowest maximum height. From the given inputs we consider the tree with the highest growth rate to proceed further. After identifying the highest growth rate bamboo tree, we need to add those trees to our priority schedules and with the combination of greedy and priority scheduling we can achieve the lowest max height of the bamboo garden.

1. **Equals: [10, 10, 10, 10, 5, 5, 5, 5, 5, 5, 5, 5]**:
   From the given growth rate of the equals, 10 is the highest growth rate tree so we consider cutting that tree first to reach to the optimal solution. Also it is required that we cut each tree at least once. So the required queue is:

   Queue = [(0, 1), (2, 3), (4, 5), (6, 7), (0, 1), (2, 3), (8, 9), (10, 11),]
   The lowest maximum height is = 40

   Failed attempts:
   queue: [(0, 1), (2, 3), (4, 5),(0, 1), (2, 3), (6, 7), (0, 1), (2, 3), (8, 9),(0, 1), (2, 3), (10, 11)]
   Max height: 60

   From the above queue it is clear that we have cut all the trees at least once and then we have approached the trees with the highest growth rate i.e., with indices 0 and 1, these things we have achieved as per priority scheduling algorithm

2. **Inequality: [98, 98, 1, 1, 1, 1]:**
   For this 98 is the highest growth rate tree so we consider cutting that tree first to proceed with our approach as mentioned above. So the required queue for inequality is as followed

   Queue = [(0, 1), (2, 3), (0, 1), (4, 5)]
   The lowest maximum height is = 196

   Failed attempts:
   Nothing, it is just so simple 😁

Submitted To

Prof. Sebastian Wild

Submitted By

Name: Balkrishna Bhatt
Student Id: 201673048

3. **Split: [100, 32, 16, 8, 4, 2, 1, 1]**:

For this 100 is the highest growth rate tree so we consider cutting that tree first to proceed with our approach as mentioned above. So the required queue for split is as followed

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)]

The lowest maximum height is = 100

Failed attempts:
queue = [(0, 1), (2, 3), (0, 1), (4, 5), (0, 1), (6, 7)]
Max height: 200
queue = [(0, 1), (0, 2), (0, 3), (0, 4), (0, 1), (0, 5), (0, 2), (0, 1), (0, 6), (0, 7)]
Max height: 128

4. **Power: [96, 54, 54, 48, 24, 18, 18, 12, 6, 6, 6, 3, 3, 2, 2, 2]:**

For this 96 is the highest growth rate tree so we consider cutting that tree first to proceed with our approach as mentioned above. So the required queue for power is as followed

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)]

The lowest maximum height is = 288

Failed attempts:
queue = [(0, 1),(2, 3),(4, 5),(0,1), (6,7),(8,9),(0,2), (3,13),(1,12),(0,11),(2,6),(3,0),(1,4),(7,10),(0,2),(1,8),(3, 9),(12,13),(14,15)]
Max height: 480
queue=[(0, 1),(2, 3),(4, 5),(0,1), (6,7),(8,9),(0,2), (3,13),(1,12),(0,11),(2,6),(3,0),(1,4),(7,10),(0,2),(1,8),(3, 9),(12,13),(0, 1),(2, 3),(14,15)]
Max height: 384

5. **Fibonacci: [55, 34, 21, 13, 8, 5, 3, 2, 1, 1]:**

For this 55 is the highest growth rate tree so we consider cutting that tree first to proceed with our approach as mentioned above. So, the required queue for Fibonacci is as follows

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),]

The lowest maximum height is = 110

Failed attempts:
queue = [(0,1), (2, 3), (4, 5), (0, 1), (6, 7), (8, 9)]
Max height: 165