Programming Puzzle: Bamboo Trimming

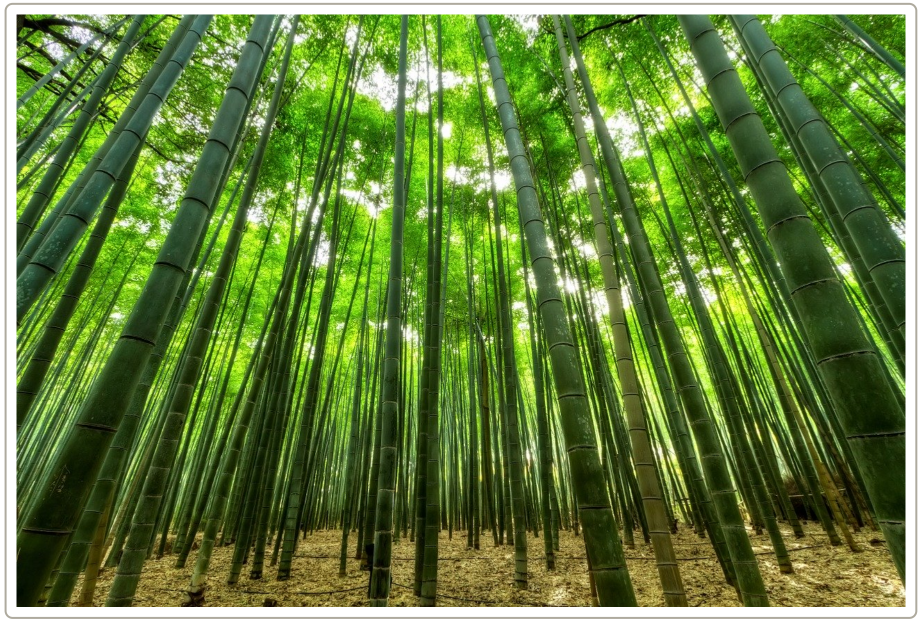
This continuous-assessment exercise consists of a small applied project with algorithmic and programming components, including a real-time leaderboard of the competition.

Will you be able to beat your classmates, or even your demonstrator?

You will be working on a real, challenging research problem, so the intention is as much on practicing the *process* of producing solutions to algorithmic problems, as on the actual deliverable.

The Bamboo Trimming Problem

To offset the long hours of sitting in classes, you and your partner are passionate gardeners, and your pride and joy is your little forest of exotic bamboos. However, being one of the fastest-growing plants on earth, the bamboo plot requires constant attention. In an attempt to keep the effort manageable, you both decide to cut down exactly one of



your the bamboo plants each day, (so 2 plants in total per day) and you cut it right back to the roots.

Since your bamboos have vastly different growth rates, some of them need more frequent cutting than others. You set out to find a periodic schedule of which bamboo to cut each day, so as to minimize the maximal height of your garden.

Formalization

You decide to mathematically model the task as follows. Given n bamboos with daily growth rates $g_1, \dots, g_n$, we assume that after growing for t days (without cutting), bamboo i will have height $t \cdot g_i$. Right after you cut a bamboo, its height is 0, and so is the initial height of all bamboos at the beginning.

Writing $h_i(t)$ for the height of bamboo i after t days, and $c_1(t)$ resp. $c_2(t)$ the bamboo that you and your partner cut on day t, we obtain:

$$h_1(0) = h_2(0) = \cdots = h_n(0) = 0;$$

$$h_i(t+1) = \begin{cases} g_i & \text{if } c_1(t) = i \vee c_2(t) = i \\ h_i(t) + g_i & \text{otherwise} \end{cases} \qquad (t \geqslant 0).$$

The task is to find an infinite schedule of cuts $c : \mathbb{N}_0 \to [n] \times [n]$ (where $c(t) = (c_1(t), c_2(t))$) that keeps the maximal height $\sup_{t \in \mathbb{N}} \max_{i \in [n]} h_i(t)$ as low as possible.

To simplify your planning, you decide to restrict your attention to *periodic* schedules: You specify a fixed, finite list C of pairs of cuts to execute, and when you are done with this list, you simply start from the beginning again, and repeat this process indefinitely.

Inputs

Your garden contains five named bamboo plots with the growth rates of the bamboos given below:

1. Equals: `[10, 10, 10, 10, 5, 5, 5, 5, 5, 5, 5, 5]`
2. Inequality: `[98, 98, 1, 1, 1, 1]`
3. Split: `[100, 32, 16, 8, 4, 2, 1, 1]`
4. Power: `[96, 54, 54, 48, 24, 18, 18, 12, 6, 6, 6, 3, 3, 2, 2, 2]`
5. Fibonacci: `[55, 34, 21, 13, 8, 5, 3, 2, 1, 1]`

Design as good a periodic schedule as you can find for each of them!

Can you argue that your solutions are best possible?

Code template

We prepared a Python implementation of the bamboo-trimming problem that you can use to evaluate your trimming schedules:

- Python sources

There is one file `bambooX.py` for each value of `X` in the list above. They simulate the growth of the bamboo under your periodic schedule and report the maximum height ever reached. The classes automatically store your results in a `csv` file. To run the simulation, extract the zip archive to a folder and run, e.g., `python3 bambooInequality.py` there.

Obey the comments! Once you downloaded the code, in each of the `bamboo*.py` files, add your periodic schedule to list `queue`.

Deliverables

This is an individual project; each student has to submit their own solution. The submission is on Canvas, split into the following parts; **you have to submit *all* of them.** Note further that only **typed solutions** will be marked, but you can use any format that Canvas accepts.

- **Part 1:** Your written report of how you arrived at your schedules
  (not more than one page)
  Describe any systematic approaches you used (if applicable) and also report on dead ends you tried (what did not work). Structure your writing with sections. A secondary criterion here is *simplicity* of the solution: If you cannot seem to

improve the heights, extra marks can be awarded for the simplest solution that achieves that solution quality.

- **Part 2:** The quality of your solutions and code
  1. smallest height you could achieve for all bamboo plots
  2. heighest lower bound for the height you could prove impossible for all bamboo plots
  3. a zip file with all your Python scripts (the `bamboo*.py` files)
- **Part 3:** Your lower-bounds proofs.
  For each lower bound from the previous question, describe why this height is impossible to achieve (by any schedule) in the given bamboo plot.
  Your answer should have one section for each bamboo plot. Each section should contain a concise proof of why this height is not achievable.

Marking

The overall mark will consist of a weighted average:

- 20% for the report.
- 50% for the quality of the achieved solutions. The baseline are solutions that Ben has found; in principle you could get more than 100% for this subtask if you manage to beat his solutions!
- 30% for the lower bound proofs.

Collaboration

This programming puzzle is an individual project, and you have to submit you own solution. In particular, the description of your solution must be a single-author document. As for all assessments, the University of Liverpool Policy on Academic Integrity applies; please refer to the Canvas page for details.

Collaboration in small groups (not more than five students) on the conceptual level (discussing ideas, not sharing entire solutions) are accepted, *but they must be declared in the description document*, including proper mention of others' contributions.

Leaderboard

We run a (voluntary, anonymous) *leaderboard* of the current best solutions. Whenever you have a periodic schedule tried in the simulator, use form below to share your achievements with the rest of the class!

Bamboo trimming leaderboard

For each of the bamboo plots listed below, you can enter the best maximal height (as output by the simulation) you were able to achieve.

bhattbalkrishna7@gmail.com (not shared)
Switch account

* Required

Your alias *

Your answer

Equals

Your answer

Inequality

Your answer

Split

Your answer

Power

Your answer

Fibonacci

Your answer

Submit Clear form

Highscores

The plots below show all answers over time. Recall that lower is better.

New submissions are immediately added at the right end, but might take a few seconds and refreshing before they show up.

	Your alias	Inequality ▲
1	Maddi Pranavi Reddy	100
2	Yinchu	196
3	Mr Theory	196
4	DailyHigh	196
5	Dominik	196
6	:)	196
7	Jinlong Liu	196
8	(͡° ͜ʖ ͡°)	196
9	Hongyuan	196
10	Z Xiong (OCT 24)	196

< > 1 2 3 4

	Your alias	Equals ▲
1	Yinchu	40
2	yarc	40
3	Mr Theory	40
4	DailyHigh	40
5	Dominik	40
6	Jinlong LIU	40
7	(͡° ͜ʖ ͡°)	40
8	Hongyuan	40
9	Z Xiong (OCT 24)	40
10	╭つ ◕◡◕ ╮つ	40

< > 1 2 3 4

	Your alias	Split ▲
1	Maddi Pranavi Reddy	98
2	Yinchu	100
3	Dominik	100
4	(͡~ ͜ʖ ͡°)	100
5	Hongyuan	100
6	Z Xiong (OCT 24)	100
7	Jinlong Liu	100
8	╭つ ◕‿◕ ╮つ	100
9	Ankita Bera	100
10	Jachym	100

‹ › 1 2 3 4

	Your alias	Power ▲
1	li3010-2	150
2	jachym	192
3	Yinchu	198
4	Hongyuan	198
5	FN	198
6	xiaojie	198
7	li3010	204
8	Dominik	216
9	Stefan	216
10	Jinlong Liu	216

‹ › 1 2 3 4

	Your alias	Fibonacci
1	Maddi Pranavi Reddy	70
2	Yinchu	105
3	FN	105
4	Adam	105
5	Yuntao Yu	105
6	Yinchu	110
7	Dominik	110
8	:)	110
9	(͡° ͜ʖ ͡°)	110
10	Jinlong Liu	110

1 2 3 4

Sebastian Wild

Contact:

- sebawild at gmail · PGP
- wild at liv.ac.uk · PGP
- ~~wild at uwaterloo.ca~~ · PGP
- ~~wild at cs.uni-kl.de~~
- Website at UoL

Elsewhere:

- GitHub profile
- LinkedIn profile
- ORCID: 0000-0002-6061-9177
- Google Scholar profile
- DBLP publication list
- Semantic Scholar Author Page

Quick links:

- my publications
- my library

- intranet site at UoL
- TCS @ Liverpool
- (Old) website TU KL
- arXiv Author ID