# The tree of program difficulty
*What makes a programming problem hard?*

Here's the rubric we followed for grading the different metrics -

| Metric | Score | Meaning |
|---|---|---|
| **Difficulty of the data structure used within the program and returned from the target function** | 1 | Operations involve the primitive data-types defined by the languages such as int, char, float etc. |
| | 2 | Arrays are used, irrespective of the dimension of the array |
| | 3 | Any other advanced data structure is used, such as trees, linked lists etc. |
| **Difficulty in deducing the algorithm** | 1 | The approach/solution is fairly common and the candidate would have been through the core concept required in this problem in an intro to programming course. |
| | 2 | The approach/solution to the problem is not straightforward and will require deliberation from the candidate to get it right. |
| | 3 | The approach/solution to the problem requires a lot of thought and exposure to multiple concepts like recursion, advanced arithmetic operations etc. at the same time. |
| **Difficulty in implementing the algorithm** | 1 | The implementation involves very basic control structures and operators involved. |
| | 2 | The implementation requires an interaction with multiple variables and one or more control structures. |
| | 3 | The implementation requires interaction with multiple variables and needs careful handling of different data dependencies and control structures. |
| **Edge information** | 1 | Does not have any specific corner cases to be thought through to get the entire logic right. |
| | 2 | Specific corner cases need to be thought through to get the entire logic right. |