

# Ad Hoc Questions

ICT and SCT Captains

October 2020

## 1 Introduction

Ad Hoc questions refer to any type of problem that cannot be categorized in a particular field, with the Latin literally translating "For This". Ad Hoc questions will have unique solutions, meaning there is not one theory or concept that can be learned to conquer all occurrences of Ad Hoc. Conversely, Ad Hoc questions can almost always be solved without needing to know a complicated algorithm - simple logic and careful thinking can solve most Ad Hoc questions.

## 2 Prevalence

In the field of competitive programming, it is important to know how much a particular topic occurs. Certain concepts have a higher importance due to their frequent appearances in contests. In this regard, Ad Hoc is one of the most important skills to master. Nearly a fifth of all questions at ICPC, IOI, or any Codeforces contest will be classified as Ad Hoc. Ad Hoc questions appear sprinkled throughout the USACO season, with Bronze Division questions being almost entirely Ad Hoc.

These questions are often the easier problems in the contest, meaning that speed is crucial to set yourself apart from other participants.

## 3 Improvement

Most Ad Hoc questions are simple, both in understanding and implementation, but don't be fooled into assuming Ad Hoc questions are easy. Many Ad Hoc questions will have edge cases and important connections that can waste a lot of time if missed. The key to improvement lies in knowing how to pick apart an Ad Hoc question. As frequently said, practice is essential, but only if you are noticing why you are struggling on a certain part of a question.

Perhaps edge cases give you a lot of trouble, particularly the  $n = 0$  case. Or maybe your issue lies in not being able to understand the problem statement. Either way, your goal should be to hone in on areas where you are weak.

Although Ad Hoc questions can't be categorized, these are frequent types of Ad Hoc questions that will appear in contests:

- Trivial Ad Hoc
- Games (Chess, Tic-Tac-Toe, Cards)
- Implementation/Brute Force
- Math/Binary
- Ad Hoc Ad Hod

## 4 Example Questions

As is the key with every programming concept, let's take a look at some questions and discuss how you should approach these problems.

- CF Round 652 B
- CF Round 646 B
- CF Round 633 B
- CF Round 618 A
- Cf Round 633 A