

ADT Selection

[Here is a video walkthrough of the solutions.](#)

Suppose we have a TA Shreyas who teaches multiple discussion sections! A student may frequent more than one discussion section. For each situation below, choose the best ADT(s) out of the following — `Map`, `Set`, `List` — and explain how you can use the ADT(s) to solve the problem. Each subpart is independent of the previous. One answer may involve multiple ADTs. There may be multiple efficient answers for each problem.

1. Storing all the `Students` in Shreyas's first section in alphabetical order.
2. Storing all the `Students` by their section, where `Students` within a section are sorted alphabetically.
3. Storing the `Students` in *all* of Shreyas's sections. There shouldn't be duplicates.
4. Quickly getting a `Student` by `sid`.
5. Quickly getting all `Students` of a given name. Names aren't necessarily unique.
6. Cycling through the `Students` in one discussion section.

Solution:

1. Put the `Students` in a `List` in alphabetical order.
2. Use a `Map`, where each `Section` maps to an alphabetically ordered `List` of `Students` in that section.
3. Use a `Set`. Add all the `Students` to the `Set`. Since a set requires elements to be unique, calling `add` on a student already in the set will not add a duplicate.
4. Use a `Map`, where each `sid` maps to one `Student`.
5. Use one `Map` that maps names to a `List` (or `Set`) of `Students` of the given name.
6. Put the `Students` in a `List`. You could use a `LinkedList` and repeatedly remove from the front and reinsert at the back. Equivalently, you could use

an ArrayList and keep an index pointer.