Which DSs are best for given ADT?

I consider that the dynamic vector is more apropriate to use for sorted list, because after you insert in corect order in dynamic list it is easy to work with the sorted list, but in binary search tree you do not have possitions and for getting a sorted order of the elements you must make in-order search wich is time consuming.

Dynamic vector is also naturally more apropriate for sorted list.

Which ADTs are best to be used to solve the given problem?

I chose the problem with balloons that move vertically and to check how many balloons do not colide. My ADT (sorted list) was perfect for this problem because I had everything sorted from the moment I stored the balloons in the ADT, so I just need to do few things, like checking if the balloons collide and after that to remove the apropriate one.

Why do we study DS? Why ADT are important? In real world of programming it is mandatory to know how to implement a framework or mabe an engine. For that we need to know how to structure everything, how to make it more secured and provide a very structured API. Data structures and algorithms provide a good understanting of how to combine libraries with other parts of the project. It learns you how to make the code more reusable and robust .