For this particular problem strategy pattern works well because we are dealing with a single function (searching), but multiple different implementations of that function are desired. Thus, it is best to create an interface which abstracts the search function so that multiple concrete implementations can inherit that functionality and differentiate it. This allows for a flexible usage of the search function, such that the search strategy can be chosen at run-time and many different implementations can be added. One drawback of this particular usage of the strategy pattern is that certain strategies will objectively be better than others, rendering those others infeasible choices and eliminating the efficacy of choosing them as a search strategy. As such, it is possible to simply implement the best strategy by relying solely on inheritance rather than creating a separate interface.