**Принципы поиска в неупорядоченном массиве.**

Надо сказать, формулировка весьма туманна. Он как-то не заострял на этом внимания.

1. Самым очевидным решением тут будет перебор всех элементов массива. Это дедовский брутфорс, зато реализуется без напряга.  
   Очевидно, что такой подход крайне не оптимален при больших объемах данных (в худшем случае придется обойти все элементы массива).
2. Решение номер 2 – деревья поиска. По сути всё очевидно: создаем дерево поиска на основе данного массива и всё. Создание дерева требует дополнительных расходов памяти, зато увеличивает эффективность поиска в разы. Дело в том, что для дерева количество операций, требуемых для достижения любого элемента <= высоте дерева. Однако тут кроется одна маленькая деталь: эффективность работы дерева напрямую зависит от его балластировки. У сбалансированного дерева максимальная глубина его правого и левого поддеревьев отличается не более чем на 1. В худшем случае одного из поддеревьев может не быть совсем (nil), а все элементы будут содержаться во втором. Тогда дерево вырождается в список (и эффективность падает соответственно). Кол-во операций для достижения элемента будет <= кол-ву элементов. Короче, в худшем случае придется перебирать все элементы.

Ну, вот и всё. Я не знаю, что тут еще можно сказать.  
(FixMe): может всунуть сюда же и сортировки? Потом посмотрим.