Нахождение в массиве пары элементов с заданной суммой

Описание задачи: для неотсортированного целочисленного массива найдите пару элементов с заданной суммой в нем.

Например,

Вход:  
   
arr = [8, 7, 2, 5, 3, 1]  
sum = 10  
   
Выход:  
 пара найдена (8, 2)  
или  
пара найдена (7, 3)  
  
Вход:  
arr = [5, 2, 6, 8, 1, 9]  
sum = 12  
  
Выход: пара не найдена

Есть несколько методов решения этой проблемы с использованием грубой силы, сортировки и хеширования. Они обсуждаются ниже:

1. Использование метода полного перебора (brute force)

Наивное решение - рассматривать каждую пару в данном массиве и возвращать, если желаемая сумма найдена. Этот подход продемонстрирован ниже на языках C, Java и Python:

Временная сложность приведенного выше решения составляет O (n2) и не требует дополнительного места, где n - размер ввода.

2. Использование сортировки

Идея состоит в том, чтобы отсортировать данный массив в порядке возрастания и сохранить пространство поиска, поддерживая два индекса (low и high), которые изначально указывают на две конечные точки массива. Затем уменьшайте пространство поиска arr[low…high] на каждой итерации цикла, сравнивая сумму элементов, присутствующих в индексах, low и high желаемую сумму. Прирост, low если сумма меньше ожидаемой; в противном случае уменьшите, high если сумма больше желаемой. Если пара найдена, верните ее.

Ниже приводится реализация C ++, Java и Python, основанная на этой идее:

Временная сложность вышеуказанного решения составляет O (n.log (n)) и не требует дополнительного места.

3. Использование хеширования

 Мы можем использовать хеш-таблицу для решения этой проблемы за линейное время. Идея состоит в том, чтобы вставить каждый элемент массива arr[i] в карту. Мы также проверяем, существует ли разница (arr[i], sum - arr[i]) на карте или нет. Если разница видна раньше, распечатайте пару и вернитесь. Алгоритм может быть реализован следующим образом на C ++, Java и Python:

Временная сложность приведенного выше решения составляет O (n) и требует дополнительного пространства O(n), где n - размер входных данных.  
Упражнение: расширите решение, чтобы вывести все пары в массиве с заданной суммой.