Understanding the problem statement

Input : arr = [1,3,4,5,6] output : we have to find the smallest number that cannot be formed by adding the numbers in the input array. start from 1 . 1 can be formed by 1 then 2 --> cannot be formed from array number. so answer is 2 # ans : 2Input : arr = [4,13,2,3,1] output : we have to find the smallest number that cannot be formed by adding the numbers in the input array. start from 1 . 1 can be formed by 1 then 2 --> 2 can be formed by 2 3 --> 3 4 --> 4 5 --> 4+1 6 --> 4+2 7 --> 4+3 8 --> 4+3+1 9 --> 4+3+2 10 --> 4+3+2+1 11 --> cannot be formed # ans : 11

Algorithm

1. Sort the array in increasing order 2. Let P be the smallest sum that we cannot make . p = 1 3. If the current number is bigger than smallest sum we cannot make so far i.e a[i] > P then P is the smallest number we cant make. return P 4. else if current number is less than or equal to smallest sum we cannot make so far i.e a[i] <= P update p as a[i]+P i.e P=a[i]+P

Complexity:

Time:

```
        sort takes O(nlogn). Steps 3 and 4 requires traversing the input
arr O(n). So total time complexity is
        O(nlogn)
```

Space:

```
        If we use inplace sorting algorithm space complexity is O(1).
Algorithm itself doesnt use any extra
        space . Space depends on sorting algorithm that we use
```

Algorithm explanation with example

input arr : 4,13,2,3,1 sort : 1,2,3,4,13 1. Let P be 1. compare with current element i.e 1 curr_ele<=P so update P to be 1+1 i.e 2 2. Again compare P with current element i.e 2 curr_ele<=P so update P to be 2+2 i.e 4 3. Again compare P with current element i.e 3 curr_ele<=P. so update P to be 4+3 i.e 7 That means 7 is the smallest sum that we cannot make with 1,2,3 4. Again compare P with current element i.e 4 curr_ele<=P so update P to be 7+4 i.e 11 That means 11 is the smallest sum that we cannot make with 1,2,3,4 5. Again compare P with current element i.e 13 curr_ele > P . so return P. our required answer is 11. 11 is the smallest sum we cannot make with 4,13,2,3,1. How is it working? ans : suppose n is the smallest sum we cannot make till now.and curr_ele is x. so we can make sums of 1,2,3...(n-1).and with x we can further make sums like x+1,x+2...x+(n-1) now if we have x

Implementation

In [4]:
```python
def findSmallestSum(arr,size):
    arr = sorted(arr)
    p = 1
    i = 0
    curr_ele = arr[i]
    while(curr_ele <=p  and i<size):
        p = p+curr_ele
        i += 1
        curr_ele = arr[i]
    return p
```

In [5]:
```python
arr = [1,3,4,5,6]
size = len(arr)
print(findSmallestSum(arr,size))
# ans 2
```

```
2
```

In [7]:
```
arr = [4,13,2,3,1]
size = len(arr)
print(findSmallestSum(arr,size))
# ans 11
```

```
11
```

In [ ]: