

**Success** Details >

Runtime: **4 ms**, faster than **100.00%** of C++ online submissions for Minimum Amount of Time to Fill Cups.

Memory Usage: **11.8 MB**, less than **100.00%** of C++ online submissions for Minimum Amount of Time to Fill Cups.

Next challenges:

Maximum Score From Removing Stones

Maximum Running Time of N Computers

Show off your acceptance:   

Time Submitted	Status	Runtime	Memory	Language
07/11/2022 12:21	Accepted	4 ms	11.8 MB	cpp
07/11/2022 12:21	Accepted	6 ms	11.8 MB	cpp

```

1 class Solution {
2 public:
3     int fillCups(vector<int>& a) {
4
5         sort(a.begin(),a.end());
6
7         int sum = 0;
8
9         int cold = a[0];
10        int warm = a[1];
11        int hot = a[2];
12
13        sum = cold + warm + hot;
14
15        if(cold + warm > hot) return sum/2+sum%2;
16
17        if(cold == 0 && warm == 0)return hot;
18        return hot;
19    }
20 };
    
```

Your previous code was restored from your local storage. [Reset to default](#)

Success Details >

Runtime: 728 ms, faster than 33.33% of Python3 online submissions for Smallest Number in Infinite Set.

Memory Usage: 14.6 MB, less than 60.00% of Python3 online submissions for Smallest Number in Infinite Set.

Next challenges:

First Missing Positive

Show off your acceptance: f t in

Time Submitted	Status	Runtime	Memory	Language
07/11/2022 12:25	Accepted	728 ms	14.6 MB	python3

```
1 class SmallestInfiniteSet:
2
3     def __init__(self):
4         self.visited = set(x for x in range(1,1001))
5
6     def popSmallest(self) -> int:
7         temp = min(self.visited)
8         self.visited.remove(temp)
9         return temp
10
11     def addBack(self, num: int) -> None:
12         if num not in self.visited:
13             self.visited.add(num)
14
15
16 # Your SmallestInfiniteSet object will be instantiated and called as such:
17 # obj = SmallestInfiniteSet()
18 # param 1 = obj.popSmallest()
19 # obj.addBack(num)
```