3. Find the Kth Smallest Sum of a Matrix With Sorted Rows

You are given an m x n matrix mat that has its rows sorted in non-decreasing order and an integer k.

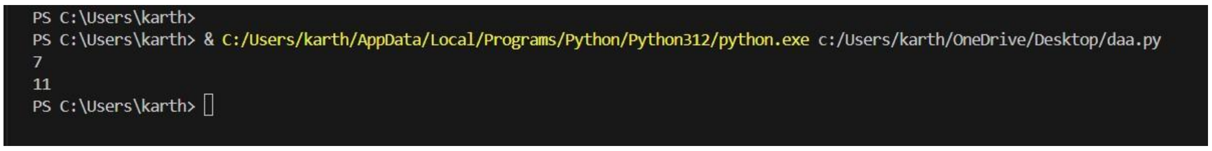
You are allowed to choose exactly one element from each row to form an array.

Return the kth smallest array sum among all possible arrays.

Code:

```
import heapq
def kthSmallest(mat,k):
    m,n=len(mat),len(mat[0])
    min_heap=[]
    initial_tuple=(sum(row[0] for row in mat), [0] * m)
    heapq.heappush(min_heap, initial_tuple)
    visited=set()
    visited.add(tuple([0]*m))
    for _ in range(k-1):
        current_sum,indices=heapq.heappop(min_heap)
        for i in range(m):
            if indices[i]+1<n:
                new_indices=indices[:]
                new_indices[i]+= 1
                new_sum=current_sum-mat[i][indices[i]]+mat[i][new_indices[i]]
                new_tuple=(new_sum, new_indices)
                if tuple(new_indices) not in visited:
                    heapq.heappush(min_heap,new_tuple)
                    visited.add(tuple(new_indices))
    return heapq.heappop(min_heap)[0]
mat=[[1, 3, 11],
    [2, 4, 6]]
k=5
print(kthSmallest(mat,k))
mat=[[1, 3, 11],
    [2, 4, 6],
   [5, 6, 7]]
k=6
print(kthSmallest(mat,k))
```

output:



```
PS C:\Users\karth>
PS C:\Users\karth> & C:/Users/karth/AppData/Local/Programs/Python/Python312/python.exe c:/Users/karth/OneDrive/Desktop/daa.py
7
11
PS C:\Users\karth> 
```

Time complexity:

F(n)=o(kmlogn)