1. Выдает ошибку не знаю \*-\*

MOD = 10\*\*9 + 7

def calculate\_mixing\_sum(n):

result = 0

i = 1

while i <= n:

j = i

while j <= n:

result += i & j

result %= MOD

j += 1

i += 1

return result

t = int(input())

for \_ in range(t):

n = int(input())

print(calculate\_mixing\_sum(n))

В этом коде функция calculate\_mixing\_sum(n) реализует алгоритм подсчета суммы смешения красок для заданного числа n. Она использует два вложенных цикла i и j, чтобы пробежать по всем парам чисел от 1 до n и прибавить результат побитового И к общей сумме. Затем результат берется по модулю 10^9 + 7 (обозначенного как MOD), чтобы гарантировать, что ответ будет соответствовать требуемому формату.

Главный цикл программы принимает входные данные, вызывает функцию calculate\_mixing\_sum(n) для каждого набора данных и выводит результат.

Пожалуйста, обратите внимание, что так как входные данные очень велики, программа может занимать некоторое время для обработки каждого из наборов тестов, и для больших значений n может потребоваться оптимизация алгоритма.

1. n, m = map(int, input().split())  
   costs = list(map(int, input().split()))  
     
   def min\_distance(n, m, costs):  
    sorted\_costs = sorted(enumerate(costs, start=1), key=lambda x: x[1])  
    selected = sorted\_costs[:m]  
    total\_cost = sum(cost for \_, cost in selected)  
     
    max\_index = max(index for index, \_ in selected)  
    min\_index = min(index for index, \_ in selected)  
    distance = max\_index - min\_index  
     
    return total\_cost + distance  
   print(min\_distance(n, m, costs))