DAA
Lab Assignment 1
Ikjot Singh
102116071

binarySearch.py

assignment1 > binarySearch.py > binarySearchIter

```python
def binarySearchRec(arr, l, r, x):
    if r >= l:
        mid = l + (r - l) // 2
        if arr[mid] == x:
            return mid
        elif arr[mid] > x:
            return binarySearchRec(arr, l, mid-1, x)
        else:
            return binarySearchRec(arr, mid + 1, r, x)
    else:
        return -1
def binarySearchIter(v, x):
    lo = 0
    hi = len(v) - 1
    while hi - lo > 1:
        mid = (hi + lo) // 2
        if v[mid] < x:
            lo = mid + 1
        else:
            hi = mid
    if v[lo] == x:
        print("Found At Index", lo)
    elif v[hi] == x:
        print("Found At Index", hi)
    else:
        print("Not Found")
```

```python
a=list(map(int,input("Enter list: ").split()))
x=int(input("enter value to be searched: "))
a.sort()
y=binarySearchRec(a,0,len(a),x)
print("By recursion binary search:",y)
x=int(input("enter value to be searched: "))
print("By iterative binary search: ")
binarySearchIter(a,x)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

```
PS C:\Users\Ikjot singh\Coding\DAA\assignment1> cd "c:/Users/Ikjot singh
PS C:\Users\Ikjot singh\Coding\DAA\assignment1> & "C:/Program Files (x86
ingh/Coding/DAA/assignment1/binarySearch.py"
Enter list: 10 25 69 42 86 98
enter value to be searched: 42
By recursion binary search: 2
enter value to be searched: 6
By iterative binary search:
Not Found
```

mergeSort.py

assignment1 > mergeSort.py > mergeIter

```python
def mergeSortRec(arr):
    if len(arr) > 1:
        mid = len(arr)//2
        L = arr[:mid]
        R = arr[mid:]
        mergeSortRec(L)
        mergeSortRec(R)
        i = j = k = 0
        while i < len(L) and j < len(R):
            if L[i] <= R[j]:
                arr[k] = L[i]
                i += 1
            else:
                arr[k] = R[j]
                j += 1
            k += 1
        while i < len(L):
            arr[k] = L[i]
            i += 1
            k += 1
        while j < len(R):
            arr[k] = R[j]
            j += 1
            k += 1

a=[10,69,2,89,420,79,500]
mergeSortRec(a)
print(a)
```

mergeSort.py

assignment1 > mergeSort.py > ...

```python
def mergeSortIter(a):
    width = 1
    n = len(a)
    while (width < n):
        l=0
        while (l < n):
            r = min(l+(width*2-1), n-1)
            m = min(l+width-1,n-1)
            mergeIter(a, l, m, r)
            l += width*2

        width *= 2
    return a
def mergeIter(a, l, m, r):
    n1 = m - l + 1
    n2 = r - m
    L = [0] * n1
    R = [0] * n2
    for i in range(0, n1):
        L[i] = a[l + i]
    for i in range(0, n2):
        R[i] = a[m + i + 1]
    i, j, k = 0, 0, l
    while i < n1 and j < n2:
        if L[i] <= R[j]:
            a[k] = L[i]
            i += 1
        else:
            a[k] = R[j]
            j += 1
```

mergeSort.py

assignment1 > mergeSort.py > mergeIter

```python
    while i < n1 and j < n2:
        if L[i] <= R[j]:
            a[k] = L[i]
            i += 1
        else:
            a[k] = R[j]
            j += 1
        k += 1
    while i < n1:
        a[k] = L[i]
        i += 1
        k += 1
    while j < n2:
        a[k] = R[j]
        j += 1
```

```python
b=[10,69,2,89,420,79,500]
mergeSortIter(b)
print(b)
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

```
PS C:\Users\Ikjot singh\Coding\DAA\assignment1> cd "c:/Users
PS C:\Users\Ikjot singh\Coding\DAA\assignment1> & "C:/Progra
ingh/Coding/DAA/assignment1/mergeSort.py"
[2, 10, 69, 79, 89, 420, 500]
[2, 10, 69, 79, 89, 420, 500]
PS C:\Users\Ikjot singh\Coding\DAA\assignment1>
```

quickSort.py

assignment1 > quickSort.py > ...

```python
def partitionRec(array, low, high):
    pivot = array[high]
    i = low - 1
    for j in range(low, high):
        if array[j] <= pivot:
            i = i + 1
            (array[i], array[j]) = (array[j], array[i])
    (array[i + 1], array[high]) = (array[high], array[i + 1])
    return i + 1
def quick_sortRec(array, low, high):
    if low < high:
        pi = partitionRec(array, low, high)
        quick_sortRec(array, low, pi - 1)
        quick_sortRec(array, pi + 1, high)

a=[10,69,2,89,420,79,500]
quick_sortRec(a,0,len(a)-1)
print(a)
```

quickSort.py

assignment1 > quickSort.py > ...

```python
def partitionIter(arr,l,h):
    i = ( l - 1 )
    x = arr[h]

    for j in range(l , h):
        if   arr[j] <= x:
            i = i+1
            arr[i],arr[j] = arr[j],arr[i]

    arr[i+1],arr[h] = arr[h],arr[i+1]
    return (i+1)
def quickSortIterative(arr,l,h):
    size = h - l + 1
    stack = [0] * (size)
    top = -1
    top = top + 1
    stack[top] = l
    top = top + 1
    stack[top] = h
    while top >= 0:
        h = stack[top]
        top = top - 1
```

PROBLEMS OUTPUT DEBUG CONSOLE TERMINAL JUPYTER GITLENS

```
PS C:\Users\Ikjot singh\Coding\DAA\assignment1> cd "c:/Users/Ikjot singh/Coding/DAA/assignment1"
PS C:\Users\Ikjot singh\Coding\DAA\assignment1> & "C:/Program Files (x86)/Microsoft Visual Studio/Shared/Python37_64/python.exe" "c
ingh/Coding/DAA/assignment1/quickSort.py"
[2, 10, 69, 79, 89, 420, 500]
[2, 10, 69, 79, 89, 420, 500]
```

quickSort.py

assignment1 > quickSort.py > quickSortIterative

```python
        h = stack[top]
        top = top - 1
        l = stack[top]
        top = top - 1
        p = partitionIter( arr, l, h )
        if p-1 > l:
            top = top + 1
            stack[top] = l
            top = top + 1
            stack[top] = p - 1
        if p+1 < h:
            top = top + 1
            stack[top] = p + 1
            top = top + 1
            stack[top] = h
b=[10,69,2,89,420,79,500]
quickSortIterative(b,0,len(b)-1)
print(b)
```

Statement | Submissions | Solution

PYTH 3

Problem

The war between CCG and ghouls is raising in Tokyo, Kaneki being king of ghouls is working hard to win this war. Touka has given a very important chest to Kaneki which will increase his ghoul army's power significantly and might win them the war. But to open the chest Kaneki must solve a puzzle that is engraved on the lock of the chest.

Kaneki has N rectangles of width W and height H. Kaneki wants to fit all these rectangles in a square, ($Rectangle\ cannot\ be\ rotated$) also he doesn't want this square to be big so that his all resources will be wasted in the square instead of war. Kaneki doesn't have time to solve this puzzle as he is too busy fighting CCG operatives so he has asked for your help, you being a brilliant programmer Kaneki is sure you will be able to solve this puzzle and might treat you to world-famous anteiku's coffee.

The first line of input will contain one integer T containing the number of test cases in the input, For the next T lines, each line contains three integers N, H, and W respectively.

You need to find the minimum size of the square that will be able to fill all n rectangles in it.

Note - Use large Data types to store variables (unsigned long long for c++ and long long for java).

###Input:

```
for i in range(int(input())):
    n,h,w=map(int,input().split())
    side=0
    l=0
    u=n*max(h,w)
    while l<=u:
        m=(u+l)//2
        if (m//h)*(m//w)>=n:
            side=m
            u=m-1
        else:
            l=m+1
    print(side)
```

0:0

Test against Custom Input

```
1
10 2 3
```


Problem

The war between CCG and ghouls is raising in Tokyo, Kaneki being king of ghouls is working hard to win this war. Touka has given a very important chest to Kaneki which will increase his ghoul army's power significantly and might win them the war. But to open the chest Kaneki must solve a puzzle that is engraved on the lock of the chest.

Kaneki has N rectangles of width W and height H. Kaneki wants to fit all these rectangles in a square, ($Rectangle\ cannot\ be\ rotated$) also he doesn't want this square to be big so that his all resources will be wasted in the square instead of war. Kaneki doesn't have time to solve this puzzle as he is too busy fighting CCG operatives so he has asked for your help, you being a brilliant programmer Kaneki is sure you will be able to solve this puzzle and might treat you to world-famous anteiku's coffee.

The first line of input will contain one integer T containing the number of test cases in the input, For the next T lines, each line contains three integers N, H, and W respectively.

You need to find the minimum size of the square that will be able to fill all n rectangles in it.

Note - Use large Data types to store variables (unsigned long long for c++ and long long for java).

###Input:

0:0

Test against Custom Input

```
1
10 2 3
```

Problem Solver Badge 19 / 50

Solve 31 more problems to get Bronze Badge

Status: ✔ Correct Answer Submission ID: 86611638

Time:
0.03s

Problem

You are given an array A of N integers. You perform this operation $N-2$ times: For each contiguous subarray of **odd size** greater than 2, you find the median of each subarray(Say medians obtained in a move are $M_1, M_2, M_3, \ldots, M_k$). In each move, you remove the first occurrence of value $min(M_1, M_2, M_3, \ldots, M_k)$ from the original array. After removing the element the array size reduces by 1 and no void spaces are left. For example, if we remove element 2 from the array $\{1, 2, 3, 4\}$, the new array will be $\{1, 3, 4\}$.

Print a single integer denoting the sum of numbers that are left in the array after performing the operations. You need to do this for T test cases.

Input Format

The first line contains T denoting the number of test cases($1 \leq T \leq 10$).

The first line of each test case contains N denoting the number of integers in the array initially($4 \leq N \leq 10^5$).

The next line contains N space seperated integers denoting $A_1, A_2, A_3, \ldots, A_N$($1 \leq A_i \leq 10^9$ for all valid i).

Input Format

The first line contains T denoting the number of test cases($1 \leq T \leq 10$).

The first line of each test case contains N denoting the number of integers in the array initially($4 \leq N \leq 10^5$).

The next line contains N space seperated integers denoting $A_1, A_2, A_3, \ldots, A_N$($1 \leq A_i \leq 10^9$ for all valid i).

Output Format

Output a single integer denoting the sum of numbers left in the array after performing the operations for each test case on a new line.

Sample Input	Sample Output
2 4 2 5 3 2	7 2

Enter your code or Upload your code as file. Save Python 3 (python 3.9.5)

```python
for i in range(int(input())):
    n=int(input())
    l=list(map(int,input().split()))
    ans=min(l)+max(l)
    print(ans)
```

1:1 vscode

1:1 vscode

Test against custom input ▼ Compile & Test code Submit code

Submission ID: 79322148 / 1 second ago

RESULT: Accepted Refer judge environment

Score	Time (sec)	Memory (KiB)	Language
0	0.68734	18840	Python 3

Problem

You are given a sequence A of length n and a number k. A number $A[l]$ is special if there exists a contiguous subarray that contains exactly k numbers that are strictly greater than $A[l]$. The specialty of a sequence is the sum of special numbers that are available in the sequence. Your task is to determine the specialty of the provided sequence.

Input format

- First line: Two numbers n and k
- Second line: n integers that represent the elements of the array

Constraints

$1 \leq k \leq n \leq 10^5$

$-10^9 \leq A[l] \leq 10^9$ for all array indices l

Input format

- First line: Two numbers n and k
- Second line: n integers that represent the elements of the array

Constraints

$1 \leq k \leq n \leq 10^5$

$-10^9 \leq A[l] \leq 10^9$ for all array indices l

Sample Input	Sample Output
5 2 4 3 2 7 8	9

Time Limit: 5
Memory Limit: 256
Source Limit:

Explanation

The beauty will be $4 + 3 + 2 = 9$.

For 4, subsequence is $[4, 3, 2, 7, 8]$

Enter your code or Upload your code as file. Save Python 3 (python 3.9.5)

```python
n,k=map(int,input().split())
A=list(map(int,input().split()))
#k count of numbers greater than a[l]
sum=0
A.sort()
for i in range(n-k):
    sum=sum+A[i]
print(sum)
```

1:1 vscode

Test against custom input ▼ Compile & Test code Submit code

Submission ID: 79322155 / 7 seconds ago

RESULT: Accepted Refer judge environment

Score	Time (sec)	Memory (KiB)	Language
0	1.73281	2	Python 3