HINTS AND RESOURCES (Set – 2)

Question-1 (Kth Sort):

Hint: Will Custom Sort help here?

Resource:

https://www.studytonight.com/cpp-programs/cpp-custom-sort-method-for-stl-pair-template-program

Sample Example:

"10101"

12 63 99 34 41 (id = 1)

75 56 23 54 32 (id = 2)

75 56 79 25 94 (id = 3)

34 48 92 45 53 (id = 4)

Explanation:

S[0] = '1', which means we need the food item which have the higher value of 1st ingredient.

New arrangement is :

75 56 23 54 32

75 56 79 25 94

34 48 92 45 53

12 63 99 34 41

S[1] = '0', which means we need the food item which have The lowest value of 2nd ingredient in previous arrangement.

New arrangement is :

75 56 23 54 32

75 56 79 25 94

34 48 92 45 53

12 63 99 34 41

S[2] = ‘1’, which means we need the food item which have The higher value of 3rd ingredient in previous arrangement.

New arrangement is :

75 56 79 25 94

75 56 23 54 32

34 48 92 45 53

12 63 99 34 41

Like this after calculating for all 5 levels of filters,

Final arrangement is:

75 56 79 25 94

75 56 23 54 32

34 48 92 45 53

12 63 99 34 41

From original arrangement we take the id’s and currently

Order of id’s are:

3 2 4 1. (answer).

Question-2 (Race):

Hint1: Speed allocation for both the times can be same?

Hint2: Can sorting help here?

Hint3: For minimum ending time you have to allocate maximum speed to maximum capacity person. But after doing this, will time be the maximum for winner?

Question-3 (Digi Puzzle):

Hint1: Can you do it with digits and binary search?

Sequence: 1 2 2 3 3 3 4 4 4 4 5 5 5 5 5 6 6 6 6 6 6 7 7 7 7 7 7 7 8 8 8 8 8 8 8 8 9 9 9 9 9 9 9 9 9 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 0 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 ……

Here, for 1-digit numbers index goes from 1 to 45

(1*1 + 2*1 + 3*1 + 4*1 + …… + 9*1 = 45).

For 2-digit numbers index goes from 46 to 9855.

(10*2 + 11*2 + 12*2 + …… + 99*2 = 9810).

Pattern: number * (number of digits in the number).

Example:

from 1 to 9 index is upto 45.

at index-46 number in the sequence will be 1,

at index-47 number in the sequence will be 0.

at index-48 number in the sequence will be 1,

at index-49 number in the sequence will be 0.

K = 200(finding the number at 200^{th} index)

Hint2: How you can apply binary search here?