Find solutions for your homework   Search

Question: The sleepy town of Pampoenfontein has been attacked by dr···

Please help using C++ program to implement the following problem. Thank you.

The sleepy town of Pampoenfontein has been attacked by dragons! A number of brave knights have ridden out to try defend the town and slay the dragons. Assume that there are n dragons and m knights. Each dragon's head has a diameter, and each knight has a height. A dragon whose head is diameter d can only be killed by a knight with height h when $d \leq h$. Furthermore, a knight can only kill at most one dragon. Given a list of dragon head diameters and list of knight heights, write a program to determine the *minimum total height* of all the knights that must be used to kill all the dragons.

Example

Imagine there are two dragon's with diameters 10 and 20, and three knights with heights 12, 22, 32. The knight of height 12 cannot kill the dragon of diameter 20, but it can kill the dragon of diameter 10. The optimal solution is to have the knight of height 12 kill the dragon of height 10 and the knight of height 22 kill the dragon of height 20 for a total height of 34. The final knight is not used at all.

Input

Input consists of two lines. The first line begins with the integer n. n integers follow, each separated by a space. These integers represent the diameters of each dragon's head. Note that these diameters are not necessarily in any order. The second line begins with the integer $m \geq n$. m integers follow, each separated by a space. These integers represent the heights of each knight. Note that these heights are not necessarily in any order.

Output

Output the mimimum total heights of all knights used to kill all the dragons.

Example Input-Output Pairs

Sample Input #1	Sample Input #2
2 10 20 3 22 12 32	3 15 12 11 5 11 20 25 5 17
Sample Output #1	**Sample Output #2**
34	48

Show transcribed image text

Expert Answer ⓘ

Anonymous answered this
11,600 answers

Was this answer helpful? 👍 0   👎 0

```cpp
// C++ program to determine the minimum total height of all the knights that must be used to kill all the
dragons

#include <iostream>

using namespace std;

int main()
{
// create pointers to arrays to store the dragons head diameter and knights height
int *knights, *dragons;
// variables to store number of knights, dragons and minimum total height
int nKnights, nDragons, minTotalHeight = 0;
bool *knightUsed; // bool array to store status of each knight(used/not used)

// read the number of dragons
cin >> nDragons;
// create an array of nDragons integers
dragons = new int[nDragons];

// loop to read each dragon head diameter
for(int i=0;i<nDragons;i++)
cin >> dragons[i];
```
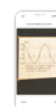
Contact us

```
knights = new int[nKnights];
// create an array of nKnights booleans
knightUsed = new bool[nKnights];

// loop to read each knight height and set status of each knight to false
for(int i=0;i<nKnights;i++){
cin >> knights[i];
knightUsed[i] = false;
}

// loop over each dragon
for(int i=0;i<nDragons;i++)
{
int idx = -1; // set index of minimum knight height to -1
// loop over the knights array
for(int j=0;j<nKnights;j++)
{
// jth knight is not used and its height is >= dragon's diameter and min index is -1 or this knight's height is
less than current minimum
if(!knightUsed[j] && (knights[j] >= dragons[i]) && (idx == -1 || knights[j] < knights[idx]))
idx = j; // update idx to j
}

// add the height of the idx knight to minTotalHeight
minTotalHeight += knights[idx];
knightUsed[idx] = true; // set idx of knightUsed to true
}

// display the minimum total height
cout << minTotalHeight;
// delete the dynamic arrays created
delete dragons;
delete knights;
delete knightUsed;

return 0;
}

// end of program
```

Output:

Run 1:

Input:

```
2 10 20
3 22 12 32
```

Output:

```
34
```

Run 2:

Input:

```
3 15 12 11
5 11 20 25 5 17
```

Output:

```
48
```

Comment ❯

Questions viewed by other students

Q: Please assist

A: See answer

Q: Problem 5: Making Change Implementation We wish to make change for an amount using as few coins as possible using an algorithm. You may use any language you choose to implement your DP change algorithm. The program should read input from a file named "amount.txt". The file contains lists of denominations (V) followed on the next line by the amount A. Example amount.txt: 1 2 5 10 1...

A: See answer   100% (1 rating)

Contact us

COMPANY ⌄

LEGAL & POLICIES ⌄

CHEGG PRODUCTS AND SERVICES ⌄

CHEGG NETWORK ⌄

CUSTOMER SERVICE ⌄

Contact us