Programy (rozwiązanie)

Autor zadania: **Karol Pokorski**
Opracowanie: **Mateusz Olszewski, Tymoteusz Wiśniewski**
Opis rozwiązania: **Lech Duraj**

Olimpiada Informatyczna Juniorów

Zadanie jest względnie proste, ale stanowi dobrą ilustrację podejścia, które umownie nazywamy *zachłannym*. Uporządkujmy (posortujmy rosnąco) zarówno wielkości programów, jak i pojemności płyt. Innymi słowy, załóżmy że $A_1 \leq A_2 \leq \ldots \leq A_N$ oraz $B_1 \leq B_2 \leq \ldots \leq B_M$. Rozważmy najmniejszy program A_1 i najmniejszą płytę B_1. Są dwie możliwości:

- $A_1 > B_1$, czyli najmniejszy program nie mieści się na najmniejszej płycie. Ale wtedy na płytę B_1 nie wejdzie też żaden inny program (bo pozostałe są takie same lub większe), a to znaczy, że płyta jest bezużyteczna. Możemy w takim razie „zapomnieć" o płycie B_1 do końca wykonywania programu i powtórzyć procedurę z programem A_1 i następną w kolejności płytą B_2.

- $A_1 \leq B_1$, co oznacza, że program A_1 da się wgrać na płytę B_1. Czy jednak to na pewno dobry pomysł? Teoretycznie mogłoby się zdarzyć, że najlepsze rozwiązanie umieszczałoby A_1 gdzie indziej, albo w ogóle pomijałoby ten program na rzecz któregoś z pozostałych. Ale zauważmy, że wtedy program A_1 wchodzi na każdą z dostępnych płyt, a więc byłoby nieopłacalne pominięcie go, albo użycie płyty B_1 do innego programu.

  Formalnie: gdyby ktoś przedstawił nam inne, „lepsze" rozwiązanie, w którym na płycie B_1 umieszcza inny program A_k, po prostu zamienilibyśmy w nim A_1 i A_k miejscami. W ten sposób otrzymalibyśmy równie dobre rozwiązanie, które A_1 nagrywa na płytę B_1. A zatem zawsze możemy postąpić zachłannie i nagrać A_1 na B_1 nie bojąc się, że „zepsujemy" optymalność rozwiązania.

  Skoro użyliśmy już zarówno najmniejszego programu, jak i najmniejszej płyty, możemy przejść do programu A_2 i płyty B_2, i powtórzyć na nich to samo rozumowanie od początku.

W ogólności, po prostu rozważamy aktualnie najmniejszy w kolejności program A_i i aktualnie najmniejszą płytę B_j. Jeśli $A_i > B_j$, to zwiększamy j o 1 („zapominamy o płycie B_j, jeśli zaś $A_i \leq B_j$, to zapamiętujemy, że nagraliśmy jeden program, po czym zwiększamy zarówno i, jak i j o 1.

Jak już zaznaczyliśmy na początku, takie podejście nazywamy *zachłannym* – w ogólności polega na robieniu tego, co wydaje się właściwe lokalnie, czyli kiedy widzimy tylko małą część danych (w tym wypadku, nagrywamy pierwszy program na pierwszą płytę, jeśli tylko jest to możliwe, nie oglądając się na resztę). Podejście zachłanne dalece nie zawsze działa, dlatego należy stosować je z rozwagą. Warto przed implementacją pomyśleć, czy jest jakiś argument za jego poprawnością.

pro.py

```
def main():
  # Wczytujemy dane - liczby N, M, długości programów i wielkości płyt.

  N = int(input())
  programy = [int(x) for x in input().split()]

  M = int(input())
  plyty = [int(x) for x in input().split()]

  # Sortowanie

  programy.sort()
  plyty.sort()

  # Właściwa pętla algorytmu

  nagrane = 0
  i = 0
  j = 0

  while i<N and j<M:          # działamy, aż skończą nam się płyty albo programy

```


Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".

```
    if programy[i]>plyty[j]: # jeśli i-ty program nie wchodzi na j-tą płytę
      j = j + 1              # zapominamy o płycie numer j

    else:                    # ale jeśli wchodzi...
      nagrane = nagrane + 1  # ...to nagrywamy i-ty program na j-tą płytę
      i = i + 1              # zapominamy o i-tym programie...
      j = j + 1              # ...oraz o j-tej płycie

  print(nagrane)

main()
```

pro.cpp

```
#include <iostream>
#include <algorithm>
#include <vector>

using namespace std;

int main() {
  // Wczytujemy dane - liczby N, M, długości programów i wielkości płyt.
  int N, M;

  cin >> N;
  vector<int> programy(N);
  for(int i=0; i<N; i++) cin >> programy[i];

  cin >> M;
  vector<int> plyty(M);
  for(int i=0; i<M; i++) cin >> plyty[i];

  // Sortowanie
  sort(programy.begin(),programy.end());
  sort(plyty.begin(),plyty.end());

  // Właściwa pętla algorytmu
  int i = 0, j = 0, nagrane = 0;

  while (i<N and j<M)  {  // działamy, aż skończą nam się płyty albo programy
    if (programy[i]>plyty[j])  // jeśli i-ty program nie wchodzi na j-tą płytę
      j++;                     // zapominamy o płycie numer j
    else {                     // ale jeśli wchodzi...
      nagrane++;               // ...to nagrywamy i-ty program na j-tą płytę
      i++;                     // zapominamy o i-tym programie...
      j++;                     // ...oraz o j-tej płycie
    }
  }

  cout << nagrane << endl;

}
```


Olimpiada finansowana jest ze środków Ministerstwa Edukacji Narodowej w ramach zadania publicznego „Organizacja i przeprowadzenie olimpiad i turniejów w latach szkolnych 2019/2020, 2020/2021, 2021/2022".