



Idealna para



Wskazówka

Wczytujemy ilość uczniów do zmiennej n oraz ich wzrosty do tablicy/listy $w[]$. Najpierw ustalamy najmniejszą różnicę wzrostów pomiędzy nimi (zmienna d) – w podwójnej pętli, gdzie znajdujemy minimalną wartość wyrażenia $|w[i] - w[j]|$. Indeks i będzie przebiegał wartości od 0 do $n - 2$, a indeks j wartości od $i + 1$ do $n - 1$. W ten sposób sprawdzone zostaną wszystkie pary uczniów (bez powtórzeń).

Następnie powtarzamy tę podwójną pętlę i zliczamy, ile razy wśród różnic wzrostów powtórza się wartość d (zmienna p).

Kod w języku C++

```
#include <bits/stdc++.h>
using namespace std;

int main()
{
    int n; cin >> n;
    int w[1000];
    for(int i = 0; i < n; i++)
        cin >> w[i];
    int d = abs(w[0] - w[1]), p = 0;
    for(int i = 0; i < n - 1; i++)
        for(int j = i + 1; j < n; j++)
            d = min(d, abs(w[i] - w[j]));
    for(int i = 0; i < n - 1; i++)
        for(int j = i + 1; j < n; j++)
            if(d == abs(w[i] - w[j])) p++;
    cout << p << endl;
}
```

Kod w języku Python

```
n = int(input())
w = [int(x) for x in input().split()]
d = abs(w[0] - w[1])
for i in range(0, n-1):
    for j in range(i+1, n):
        d = min(d, abs(w[i] - w[j]))
```

```

p = 0
for i in range(0, n-1):
    for j in range(i+1, n):
        if abs(w[i] - w[j]) == d:
            p += 1
print(p)

```

Uwagi

Działanie programu można by znacząco przyspieszyć, gdyby udało się wyeliminować podwójną pętlę, powodującą kwadratową złożoność algorytmu ($O(n^2)$). Wzrosty można najpierw posortować przez zliczanie (w czasie liniowym). Jeśli choć jedna wartość wzrostu się powtarza, wtedy $d = 0$ i tylko takie pary uczniów wchodzi w rachubę (choć być może pochodzące z różnych podzbiorów). W przeciwnym razie przeglądamy sąsiednie liczniki o niezerowych wartościach i zliczamy (w pomocniczej tablicy $A[]$), ile jest par o najmniejszej różnicy wzrostów.

Oto przykładowy kod w języku Python:

```

n = int(input())
L = [0] * 251
d0 = False
for x in input().split():
    i = int(x)
    L[i] += 1;
    if L[i] > 1:
        d0 = True
if d0:
    p = sum(el*(el-1)//2 for el in L)
else:
    i = 100
    while L[i] == 0:
        i += 1
    j = i + 1
    A = [0] * 151
    while j <= 250:
        if L[j] == 0:
            j += 1
        else:
            A[j - i] += 1
            i = j
            j = j + 1
    i = 0;
    while A[i] == 0:
        i += 1
    p = A[i]
print(p)

```