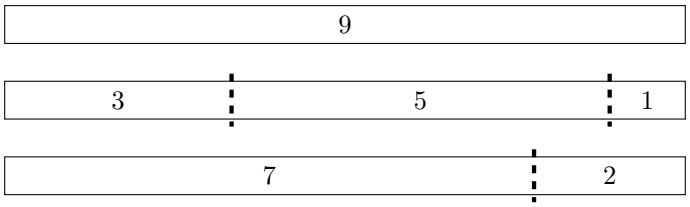
Deski kontratakują (rozwiązanie)

XIV OIJ, zawody I stopnia, tura ukryta
14 października 2019 – 13 stycznia 2020

Autorzy zadania: **Bartosz Łukasiewicz, Jacek Tomasiewicz**
Opracowanie: **Dominik Klemba, Tymoteusz Wiśniewski**

Olimpiada Informatyczna Juniorów

Zadanie to jest bardzo zbliżone do zadania z tury otwartej. Jedyną różnicą tutaj jest fakt, że w tym zadaniu deski możemy przecinać na mniejsze kawałki, a nie tylko skracać. Musimy jednak pamiętać o tym, że wszystkie długości otrzymanych desek mają być liczbami całkowitymi. Dla przykładu, poniższą deskę o rozmiarze 9 możemy podzielić na trzy mniejsze deski o rozmiarach 3, 5 oraz 1, a także na dwie deski o rozmiarach 7 i 2.



Najpierw zastanówmy się, kiedy nie da się wybudować żadnej piaskownicy – ma to miejsce tylko wtedy, kiedy nie możemy w ogóle wyciąć czterech desek, nawet o długości 1. A tak dzieje się jedynie, kiedy suma długości wszystkich desek jest mniejsza od 4 – w każdym innym przypadku możemy w najgorszym razie pociąć deski na kawałki długości 1 i skonstruować z nich małą piaskownicę.

Aby rozwiązać to zadanie zauważmy, że nigdy nie możemy użyć więcej czterech desek, jako że każdy z boków piaskownicy musi być wycięty z jednej deski. Rozważymy zatem cztery możliwości: zrobienie piaskownicy z dokładnie jednej, z dokładnie dwóch, trzech lub czterech desek. Dla każdego z tych scenariuszy znajdziemy największy możliwy do wycięcia bok (jako że szukany wynik – pole piaskownicy – jest po prostu kwadratem tej liczby) i wybierzemy najlepszą możliwość.

Jeżeli chcemy wykorzystać tylko jedną deskę, musimy z niej wykroić cztery kawałki równej długości. Oczywiście opłaca się wziąć najdłuższą deskę – używając krótszej możemy tylko stracić na długości. Jeśli największa deska ma długość d, to powstałe cztery kawałki będą miały długość $\left\lfloor \frac{d}{4} \right\rfloor$, czyli $\frac{d}{4}$ zaokrąglone w dół – na przykład jeśli $d = 9$, to możemy otrzymać cztery deski o długości $\left\lfloor \frac{9}{4} \right\rfloor = 2$. Dzielenie z zaokrąglaniem w dół (inaczej *dzielenie całkowitoliczbowe*) w języku C++ wykonujemy poprzez użycie operatora `/` (np. `d/4`), a w Pythonie – operatora `//` (np. `d // 4`).

Inną opcją jest użycie dwóch desek. Znowu łatwo zauważyć, że powinniśmy wtedy wykorzystać dwie najdłuższe deski. Oznaczmy ich długości poprzez d_1 oraz d_2 ($d_1 \geq d_2$). Tutaj mamy dwie możliwości do sprawdzenia:

- Z pierwszej deski (dłuższej) wycinamy trzy kawałki, a z drugiej jeden. Wtedy bokiem naszej piaskownicy będzie mniejsza z liczb $\left\lfloor \frac{d_1}{3} \right\rfloor$ oraz d_2. Na przykład z desek 12 i 5 otrzymamy w ten sposób piaskownicę o boku $12/3 = 4$ (deskę 5 musimy przyciąć do 4), ale z desek 24 i 7 – piaskownicę o boku 7 (z 24 moglibyśmy wykroić trzy deski nawet długości 8, ale i tak potem musielibyśmy je przyciąć do ostatniej).
- Każdą z obu desek dzielimy na dwie mniejsze deski. Wtedy długością boku naszej kwadratowej piaskownicy będzie $\min\left(\left\lfloor \frac{d_1}{2} \right\rfloor, \left\lfloor \frac{d_2}{2} \right\rfloor\right)$. Warto zauważyć, że ponieważ $d_1 \geq d_2$, to wynikiem jest po prostu $\left\lfloor \frac{d_2}{2} \right\rfloor$.

Teraz sprawdźmy, jaki bok można uzyskać za pomocą trzech desek. Ponownie bierzemy trzy najdłuższe deski $d_1 \geq d_2 \geq d_3$. Tutaj musimy wykonać jedynie jedno cięcie: deskę o długości d_1 dzielimy na dwie mniejsze. Bokiem naszej piaskownicy jest wtedy: $\min\left(\left\lfloor \frac{d_1}{2} \right\rfloor, d_2, d_3\right)$. Możemy ponownie pominąć środkowy element (d_2), ponieważ $d_2 \geq d_3$.

Ostatecznie, możemy użyć czterech desek. Weźmy cztery najdłuższe deski: $d_1 \geq d_2 \geq d_3 \geq d_4$. Wtedy bokiem naszej kwadratowej piaskownicy jest $\min(d_1, d_2, d_3, d_4) = d_4$.

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".

dek.cpp

```cpp
#include "bits/stdc++.h"

using namespace std;

int main() {
  ios_base::sync_with_stdio(0);

  // Wczytujemy liczbę desek.
  int N;
  cin >> N;

  // Deklarujemy wektor L na długości desek o długości N
  vector <int> L(N);
  // i wczytujemy wartości po kolei do tego wektora.
  for (int i=0; i<N; i++) cin >> L[i];

  // Sortujemy wszystkie długości korzystając z biblioteki stanardowej.
  sort(L.begin(), L.end());

  // W poniższych wierszach szukamy długości boku naszej kwadratowej piaskownicy.

  // Pierwszym możliwym rozwiązaniem jest wzięcie najdłuższej deski
  // i podzielenie jej na 4 równe kawałki.
  int bok = L[N-1]/4;

  // Kiedy mamy co najmniej dwie deski mamy dwa przypadki do rozpatrzenia:
  if (N >= 2) {
    // albo bierzemy najdłuższą deskę, dzielimy na trzy części i jako czwartą
    // deskę bierzemy drugą najdłuższą,
    bok = max(bok, min(L[N-1]/3, L[N-2]));
    // albo obie dwie najdłuższe deski dzielimy na połowy, otrzymując cztery deski.
    bok = max(bok, L[N-2]/2);
  }

  // Jeżeli mamy co najmniej trzy deski,
  if (N >= 3) {
    // to jedyną możliwością, którą musimy rozważyć jest przepiłowanie
    // najdłuższą deskę na pół.
    bok = max(bok, min(L[N-1]/2, L[N-3]));
  }

  // Natomiast jeżeli mamy co najmniej cztery deski,
  if (N >= 4) {
    // to możemy wziąć wszystkie deski bez żadnego piłowania.
    bok = max(bok, L[N-4]);
  }

  // Wypisujemy wynik - pole największej piaskownicy,
  // pamiętając że wynik należy zrzutować na zmienną typu long long.
  cout << (long long)bok * bok << "\n";
}
```


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".

dek.py

```python
# Wczytujemy liczbę desek.
N = int(input())

# Wczytujemy długości desek.  Wszystkie elementy konwertujemy
# z napisów na liczby.
L = [int(l) for l in input().split()]

# Sortujemy długości desek używając funkcji bibliotecznej.
L.sort()

# W poniższych wierszach szukamy długości boku naszej kwadratowej piaskownicy.

# Pierwszym możliwym rozwiązaniem jest wzięcie najdłuższej deski
# i podzielenie jej na 4 równe kawałki.
bok = L[-1]//4;

# Kiedy mamy co najmniej dwie deski mamy dwa przypadki do rozpatrzenia:
if N >= 2:
  # albo bierzemy najdłuższą deskę, dzielimy na trzy części i jako czwartą
  # deskę bierzemy drugą najdłuższą,
  bok = max(bok, min(L[-1]//3, L[-2]))
  # albo obie dwie najdłuższe deski dzielimy na połowy, otrzymując cztery deski.
  bok = max(bok, L[-2]//2)

# Jeżeli mamy co najmniej trzy deski,
if N >= 3:
  # to jedyną możliwością, którą musimy rozważyć jest przepiłowanie
  # najdłuższej deskę na pół.
  bok = max(bok, min(L[-1]//2, L[-3]));

# Natomiast jeżeli mamy co najmniej cztery deski,
if N >= 4:
  # to możemy wziąć wszystkie deski bez żadnego piłowania.
  bok = max(bok, L[-4]);

# Wypisujemy wynik - pole największej piaskownicy,
print(bok * bok)
```


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".