

Formalne przedstawienie problemu

Dane wejściowe:

Ciąg n liczb

Wynik:

Wartość, która występuje w tym zbiorze więcej niż połowę razy lub informacja o braku takiej wartości.

Rozwiązanie

Podstawowym pomysłem na rozwiązanie zadania jest sprawdzenie dla każdego elementu ciągu, czy jest on liderem tego ciągu. Takie rozwiązanie ma złożoność kwadratową, zdecydowanie zbyt wolną.

Kolejnym pomysłem byłoby posortować ów zbiór liczb na przykład rosnąco. W posortowanym ciągu wszystkie wystąpienia danego elementu znajdują się obok siebie, więc możemy zliczać kolejne pakiety sąsiednich równych elementów w ciągu posortowanym. Ten krok można już wykonać w złożoności czasowej liniowej (plus oczywiście czas, którego wymagało sortowanie, przynajmniej $n \log n$).

Okazuje się, że **da się znaleźć lidera w czasie liniowym** bez uprzedniego sortowania zbioru. Algorytm taki opiera się na podstawowym spostrzeżeniu, że jeśli zbiór liczb posiada pewnego lidera, to możemy z tego zbioru wykreślić dwa elementy **różne od siebie**, a pozostały zbiór również ma tego samego lidera. Jest tak, gdyż jeżeli z ciągu usuwamy **dwa różne** elementy, to co najwyżej jednym z nich jest lider, liczebność liderów zmalała więc najwyżej o jeden a liczebność zbioru o dwa, więc liderów nadal jest więcej niż połowa. Wykreślanie elementów z tablicy nie musi być przy tym jednoznaczne z ich usuwaniem z tej tablicy (wiemy, że usuwanie elementów z tablicy jest dość obciążające czasowo). Wystarczy jeśli będziemy postępować według poniższego schematu:

- niech pierwszy element w zbiorze zostanie oznaczony jako kandydat na lidera (zmienna o nazwie **KL**). Oznaczmy też w każdym kroku ile sztuk tych kandydatów na lidera mamy. Na początek, kiedy jest nowy kandydat na lidera, jest go dokładnie jedna sztuka (**ile=1**)
- Dla każdego kolejnego elementu z tablicy, pobierz kolejny element i sprawdź:
 - jeśli jest on różny od kandydata na lidera, to skreśl go z tablicy do pary z jednym z wystąpień KL, czyli zmniejsz o jeden wartość ile (**ile--**) (zauważ że zmienna ile wskazuje na ilość kandydatów na lidera jaka jest dostępna do skreślenia w parze z elementami różnymi od KL)
 - jeśli jest on równy kandydatowi na lidera, to zwiększ ile o jeden (**ile++**). Jest teraz o jeden więcej tych kandydatów na lidera i trzeba będzie poszukać o jeden więcej elementów różnych od niego, by je wszystkie skreślić
 - UWAGA! sprawdź czy ile wynosi zero (sytuacja taka oznacza, że właśnie skreśliliśmy wszystkich kandydatów na lidera z elementami różnymi od nich). Jeśli tak, to następny pobrany element z tablicy będzie nowym kandydatem na lidera, a jego liczebność zostanie na nowo ustalona na jeden (ile=1)
- teraz wnioski: jeśli ile wynosi po tej pętli zero, to znaczy, że w zbiorze nie było lidera (wszystkie wystąpienia każdego kandydata na lidera dały się do pary skreślić z elementami różnymi od nich). Jeśli natomiast ile jest liczbą większą od zera, to KL musi zostać zweryfikowany. To znaczy przelicz jeszcze ile razy w Twojej tablicy występuje KL. Jeśli więcej niż połowę razy to jest to rzeczywisty lider, a jeśli mniej, to w tablicy nie ma lidera.

Implementacja

```
int main()
{
    int n,t[100], ile=0,KL;
    cin >> n;
    for (int i=0;i<n; i++) cin >> t[i];
    for (int i=0;i<n;i++)
    {
        if (ile == 0) { KL=t[i]; ile=1;}
        else
            if (t[i] == KL) ile++;
            else ile--;
    }
    if (ile==0) cout << "Brak lidera";
    else
    {
        ile=0;
        for (int i=0;i<n;i++) if (t[i] == KL) ile++;
        if (ile>n/2) cout << "Liderem jest "<< KL;
        else cout << "Brak lidera";
    }
}
```