

# 1 Algorytmy alokacji mandatów

---

**Algorithm 1:** Metoda Sainte-Laguë do alokacji miejsc

---

**Data:** Lista głosów *votes*, liczba miejsc do przydzielenia *seats*

**Result:** Lista przydzielonych miejsc dla każdej partii

```
1 Zainicjuj allocated_seats jako tablice zer, o tej samej długości co votes
2 Zainicjuj heap jako pusta listę

   // Inicjalizacja kopca wartościami początkowymi dla każdej
   // partii
3 foreach indeks i i głos vote w votes do
4   | Oblicz wartość początkowa  $val \leftarrow -\frac{vote}{1.4}$            // Negatywne dla
   | zachowania kopca maksymalnego
5   | Wstaw (val, i) do heap

   // Przydzielanie miejsc
6 for j = 1 to seats do
7   | (_, party) ← Zdejmij górny element z heap
8   | allocated_seats[party] ← allocated_seats[party] + 1
   | // Oblicz ponownie wartość na podstawie aktualnie
   | przydzielonych miejsc i włóż z powrotem do kopca
9   |  $new\_val \leftarrow -\frac{votes[party]}{2 \times allocated\_seats[party] + 1}$ 
10  | Wstaw (new_val, party) z powrotem do heap

11 return allocated_seats
```

---

---

**Algorithm 2:** Metoda D'Hondta do alokacji miejsc

---

**Data:** Lista głosów *votes*, liczba miejsc do przydzielenia *seats*

**Result:** Lista przydzielonych miejsc dla każdej partii

```
1 Zainicjuj allocated_seats jako tablice zer, o tej samej długości co votes
2 Zainicjuj heap jako pusta listę i zorganizuj jako kopiec

  // Inicjalizacja kopca wartościami początkowymi dla każdej
  partii
3 foreach indeks i i głos vote w votes do
4   Oblicz wartość początkowa  $val \leftarrow -vote$  // Negatywne dla
   zachowania kopca maksymalnego
5   Wstaw (val, i) do heap

  // Przydzielanie miejsc
6 for  $j = 1$  to seats do
7   (_, party)  $\leftarrow$  Zdejmij górny element z heap
8    $allocated\_seats[party] \leftarrow allocated\_seats[party] + 1$ 
   // Oblicz ponownie wartość na podstawie aktualnie
   przydzielonych miejsc i włóż z powrotem do kopca
9    $new\_val \leftarrow -\frac{votes[party]}{allocated\_seats[party]+1}$ 
10  Wstaw (new_val, party) z powrotem do heap

11 return allocated_seats
```

---

---

**Algorithm 3:** Metoda Hare'a-Niemeyera do alokacji miejsc

---

**Data:** Lista głosów *votes*, liczba miejsc do przydzielenia *seats*

**Result:** Lista przydzielonych miejsc dla każdej partii

```
1  $total\_votes \leftarrow$  suma wszystkich wartości w votes
2  $hare\_quota \leftarrow total\_votes$  podzielone przez seats

3 Zainicjuj allocated_seats jako listę, w której każdy element jest wynikiem
  dzielenia całkowitego głosu przez hare_quota dla każdej partii
4  $remaining\_seats \leftarrow seats$  minus suma allocated_seats

5 Zainicjuj remainders jako listę, w której każdy element jest reszta z
  dzielenia głosu przez hare_quota dla każdej partii

6 while remaining_seats jest większe od 0 do
7    $largest\_index \leftarrow$  indeks największej wartości w remainders
8   Inkrementuj  $allocated\_seats[largest\_index]$  o 1
9   Dekrementuj remaining_seats o 1
10  Ustaw  $remainders[largest\_index]$  na 0
   // Zapobiegaj ponownemu przydzieleniu tej samej partii

11 return allocated_seats
```

---