

# DUOMENŲ RIKIAVIMAS

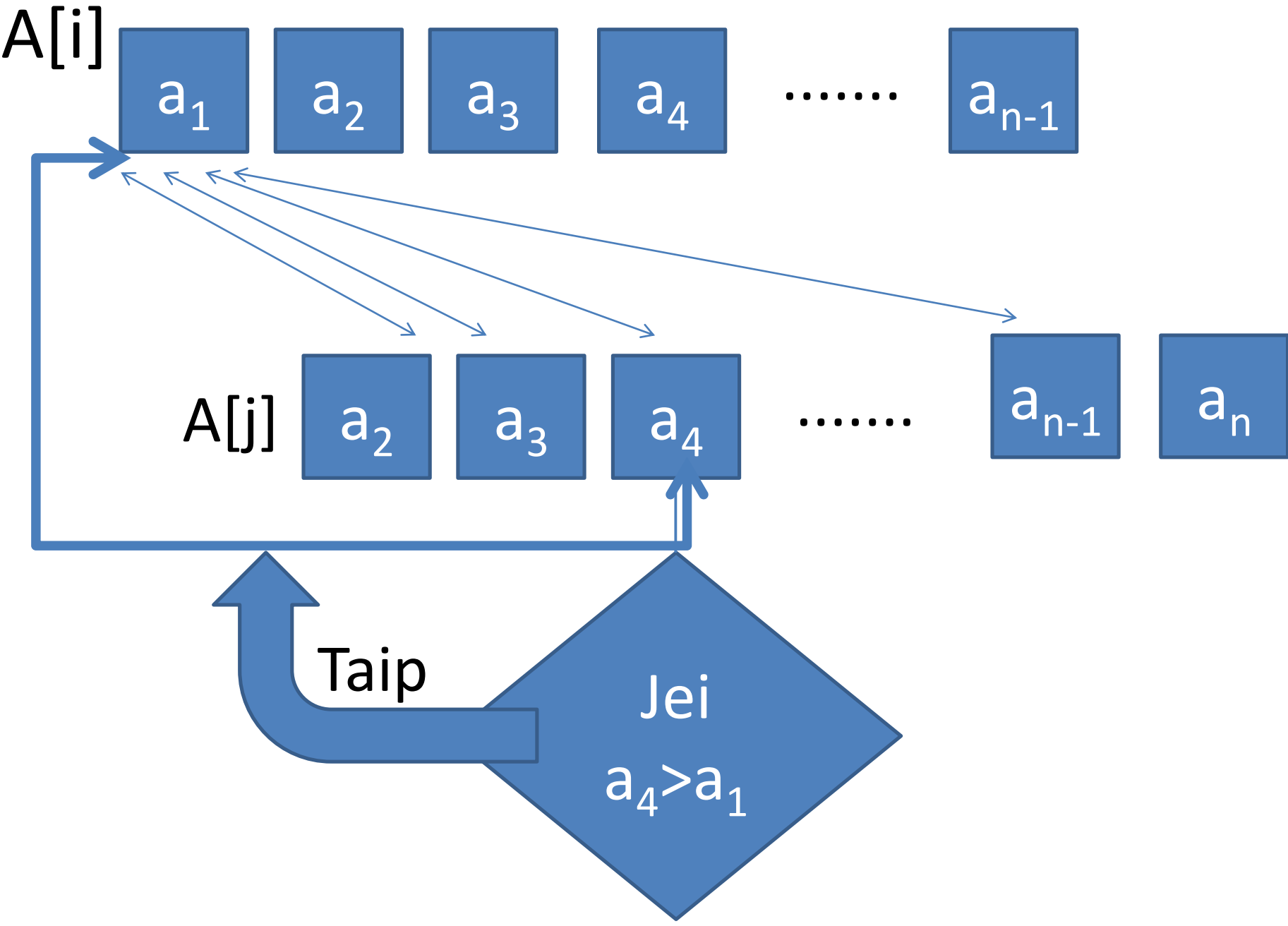
# Duomenų rikiavimo algoritmai

- **Minmax (intervalinis)**
- **Burbulas (porinių sukeitimų)**
- **Dalyba pusiau (mišrus)**
- **Pilnas minmax (intervalinis)**
- **Įterpimu (intarpų)**

**Vienas paprasčiausių ir populiariausių  
algoritmų yra rikiavimas išrinkimo  
metodu (MinMax)**

# Žodinis algoritmas

- Masyve tarp elementų  $a_1, a_2, a_3, \dots, a_n$  ieškoma didesnio už  $a_1$  ir radus jis sukeičiamas vietomis su pirmuoju  $a_1$ .
- Toliau masyve tarp elementų  $a_2, a_3, a_4, \dots, a_n$  ieškoma didesnio už  $a_2$  ir radus jis sukeičiamas vietomis su antruoju  $a_2$ . (pirmuoju paieškos intervale).
- Toliau masyve tarp elementų  $a_3, a_4, a_5, \dots, a_n$  ieškoma didesnio už  $a_3$  ir radus jis sukeičiamas vietomis su trečiuoju  $a_3$ . (pirmuoju paieškos intervale).
- Veiksmai kartojami tol kol liks vienas elementas.



**Pradžia MinMaxB (double A[], int n)**

**Kintamieji double c; int i, j;**

**i = 1, n-1**

**j = i + 1, n**

**Taip**

**A[j] > A[i]**

**Ne**

**c = A[i];**

**A[i] = A[j];**

**A[j] = c;**

**Pabaiga**

# MinMax metodas

```
void MinMaxB(double A[], int n)
{
double c; // Pagalbinis kintamasis elementų sukeitimui
for (int i = 1; i <= n - 1; i++)
    for (int j = i + 1; j <= n; j++)
        if (A[j] > A[i]) //rikiavimo raktas
        {
            c = A[i];
            A[i] = A[j];
            A[j] = c;
        }
}
```

Masyvo elementų sukeitimui galima naudoti funkciją **swap (A[j], A[i])**