



# Odd-even Transposition Sort Algorithm

⇒ Implementação **sequencial**

```
int phase, i, temp;

for (phase = 0; phase < n; phase++)
    if (phase % 2 == 0) { /* Even phase */
        for (i = 1; i < n; i += 2)
            if (a[i-1] > a[i]) {
                temp = a[i];
                a[i] = a[i-1];
                a[i-1] = temp;
            }
    }
    else { /* Odd phase */
        for (i = 1; i < n-1; i += 2)
            if (a[i] > a[i+1]) {
                temp = a[i];
                a[i] = a[i+1];
                a[i+1] = temp;
            }
    }
} /* Odd_even_sort */
```

Even phase

Odd phase



# *Odd-even Transposition Sort Algorithm*

## ⇒ Implementação **paralela**

```
public static void main(String[] args) {
    for(int fase=comeco; fase<fim; fase++) {
        if(fase%2 == 0) { //par
            ThreadEven par = new ThreadEven();
            par.start();
        } //fim do if
        if(fase%2 != 0) { //impar
            ThreadOdd impar = new ThreadOdd();
            impar.start();
        } //fim do if
    } //fim do for
} //fim do metodo main
```

//OBS: as threads par e impar contem o codigo de comparacao e (eventual) troca dos elementos