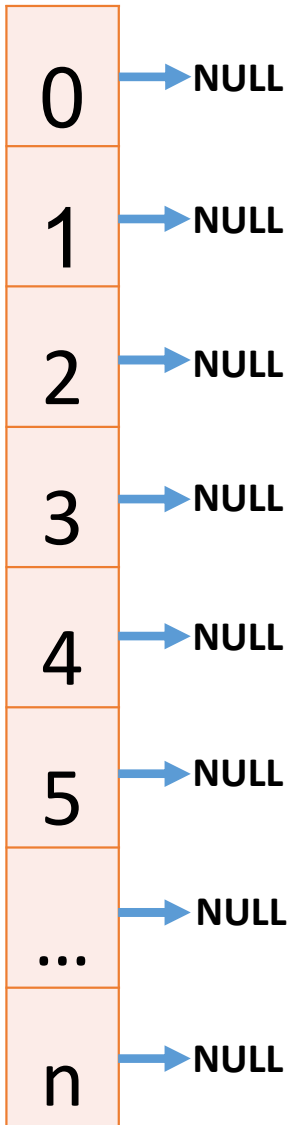


## 20\_12\_2021 Zadania numer 2 GL05

Napisz funkcję „Zadanie\_1” znajdującą element „x” w posortowanej tablicy w jak najszybszy sposób. Funkcja zwraca czy dany element znajduje się w ciągu i na której jest pozycji.

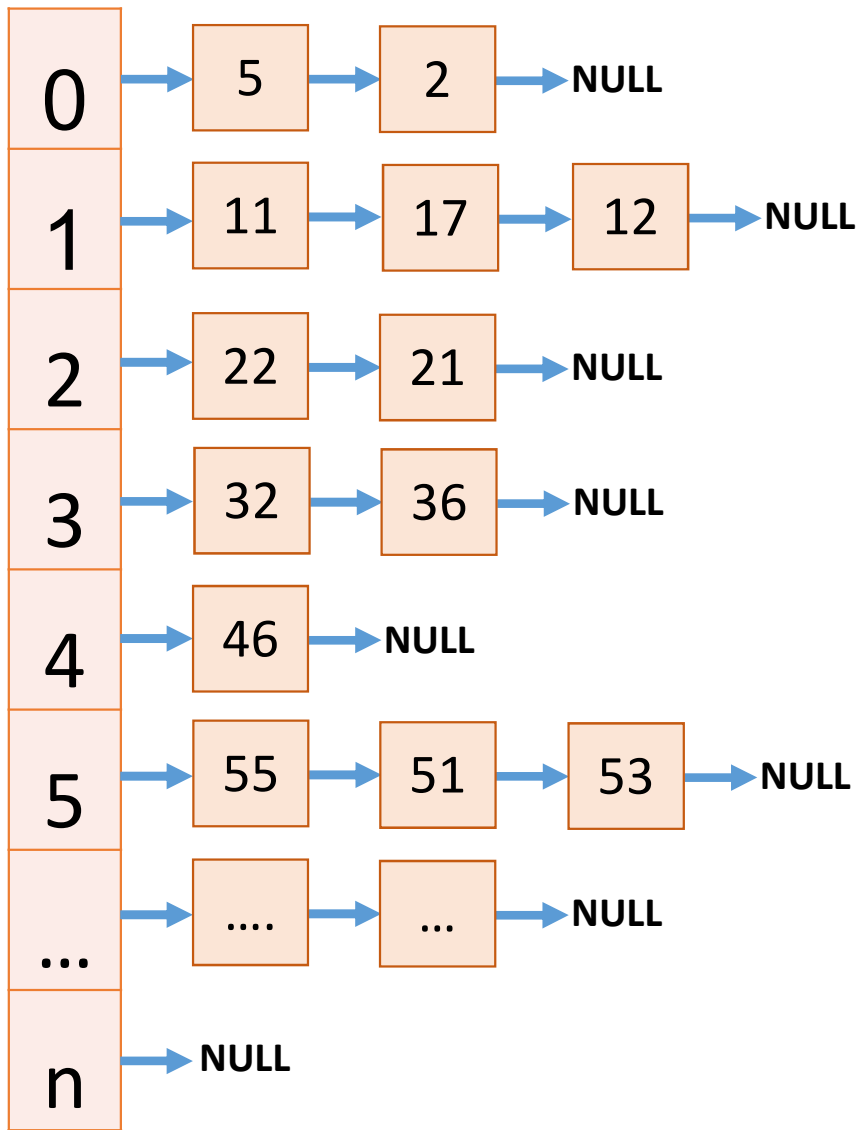
Napisz funkcję „Zadanie\_2” realizującą następujące sortowanie.



T[11,5,22,17,32, 55, 46, 2,...,36]

```
node** LN = new node* [n];  
for (int i = 0; i < n; i++)  
    LN[i] = NULL;
```

```
double Zadanie_2(int *T, int n, int BucketSize=10)  
{  
  
    return SortTime  
}
```



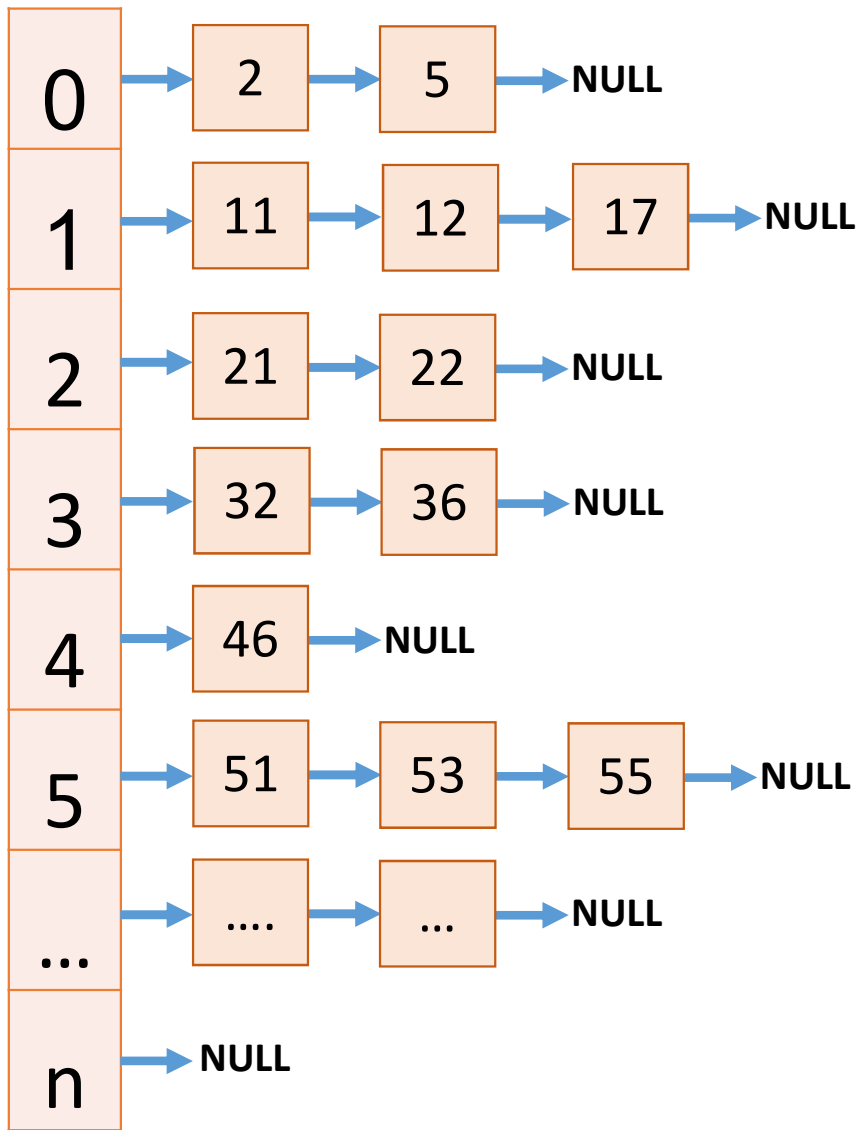
## Lista sąsiedztwa grafu

T[11, 5, 22, 17, 32, 55, 46, 2, 21, 51, 53, 12..., 36]

```

node** LN = new node* [n];
for (int i = 0; i < n; i++)
    LN[i] = NULL;

double Sortowanie(int *T, int n, int BucketSize=10)
{
    return SortTime
}
  
```



## Lista sąsiedztwa grafu

T[11, 5, 22, 17, 32, 55, 46, 2, 21, 51, 53, 12..., 36]

```

node** LN = new node* [n];
for (int i = 0; i < n; i++)
    LN[i] = NULL;
  
```

```

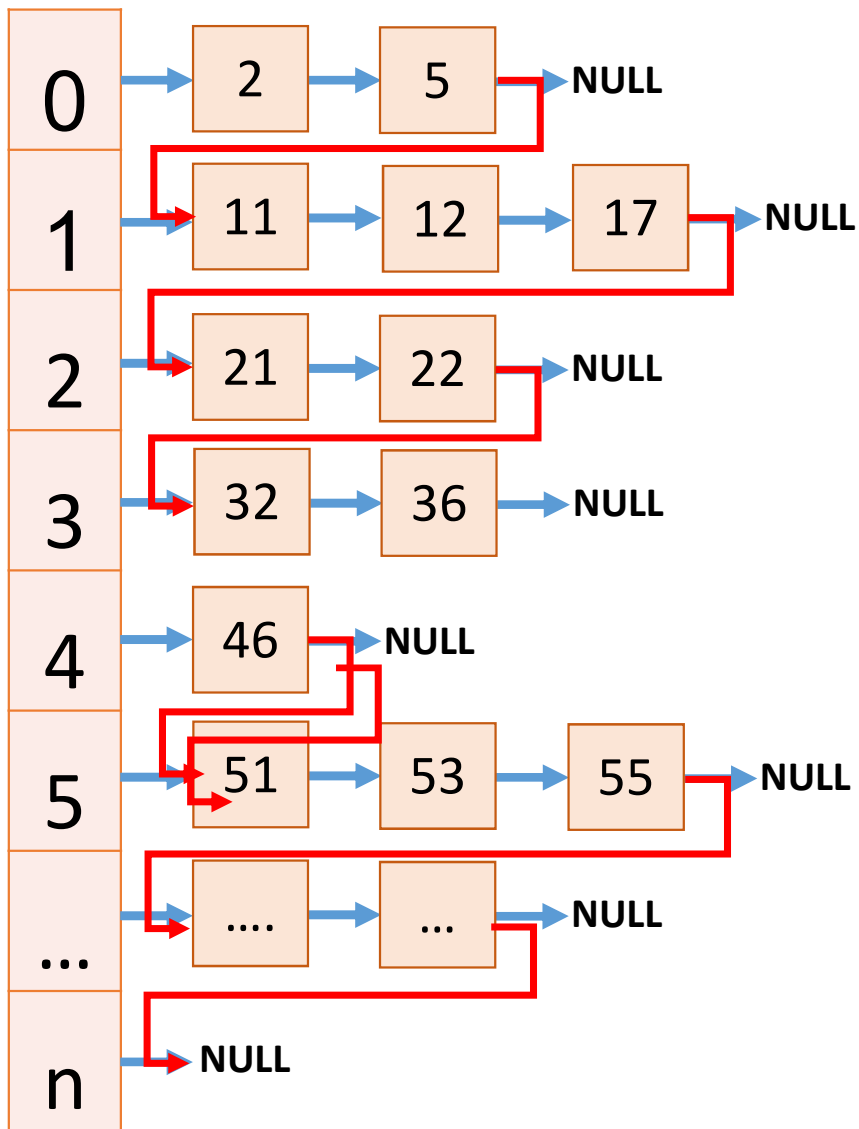
double Sortowanie(int *T, int n, int BucketSize=10)
{
  
```

```

    return SortTime
  
```

```

}
Sortowanie każdej listy
sortowaniem
bąbelkowym
  
```



## Lista sąsiedztwa grafu

T[11, 5, 22, 17, 32, 55, 46, 2, 21, 51, 53, 12..., 36]

```
node** LN = new node* [n];
for (int i = 0; i < n; i++)
    LN[i] = NULL;
```

```
double Sortowanie(int *T, int n, int BucketSize=10)
{
    return SortTime
}
```

Sortowanie każdej listy sortowaniem bąbelkowym

Łączenie posortowanych list

Wyświetlenie czasu działania