

Quick sort implementacja

Wpisany przez Ilona Simek

wtorek, 22 października 2019 20:55 - Poprawiony poniedziałek, 16 grudnia 2019 23:01

```
#include<iostream>
using namespace std;
```

```
void quick_sort(int *tab, int lewy, int prawy)
{
    if(prawy <= lewy) return;
```

```
    int i = lewy - 1, j = prawy + 1,
        pivot = tab[(lewy+prawy)/2]; //wybieramy punkt odniesienia
```

```
    while(1)
    {
        //szukam elementu wiekszego lub rownego pivot stojacego
        //po prawej stronie wartosci pivot
        while(pivot>tab[++i]);
```

```
        //szukam elementu mniejszego lub rownego pivot stojacego
        //po lewej stronie wartosci pivot
        while(pivot<tab[--j]);
```

```
        //jesli liczniki sie nie minely to zamień elementy ze soba
        //stojace po niewlasciwej stronie elementu pivot
        if( i <= j)
            //funkcja swap zamienia wartosciami tab[i] z tab[j]
            swap(tab[i],tab[j]);
        else
            break;
    }
```

```
    if(j > lewy)
        quick_sort(tab, lewy, j);
```

Quick sort implementacja

Wpisany przez Ilona Simek

wtorek, 22 października 2019 20:55 - Poprawiony poniedziałek, 16 grudnia 2019 23:01

```
if(i < prawy)
    quick_sort(tab, i, prawy);
}
int main()
{
    int *tab, n;

    cout<<"Ile liczb chcesz posortowac? ";
    cin>>n;
    tab = new int [n]; //przydzielenie pamięci na elementy tablicy
    //wczytanie liczb
    for(int i=0;i<n;i++)
        cin>>tab[i];

    quick_sort(tab,0, n-1);

    //wypisanie posortowanych elementów
    for(int i=0;i<n;i++)
        cout<<tab[i]<<" ";

    return 0;
}
```