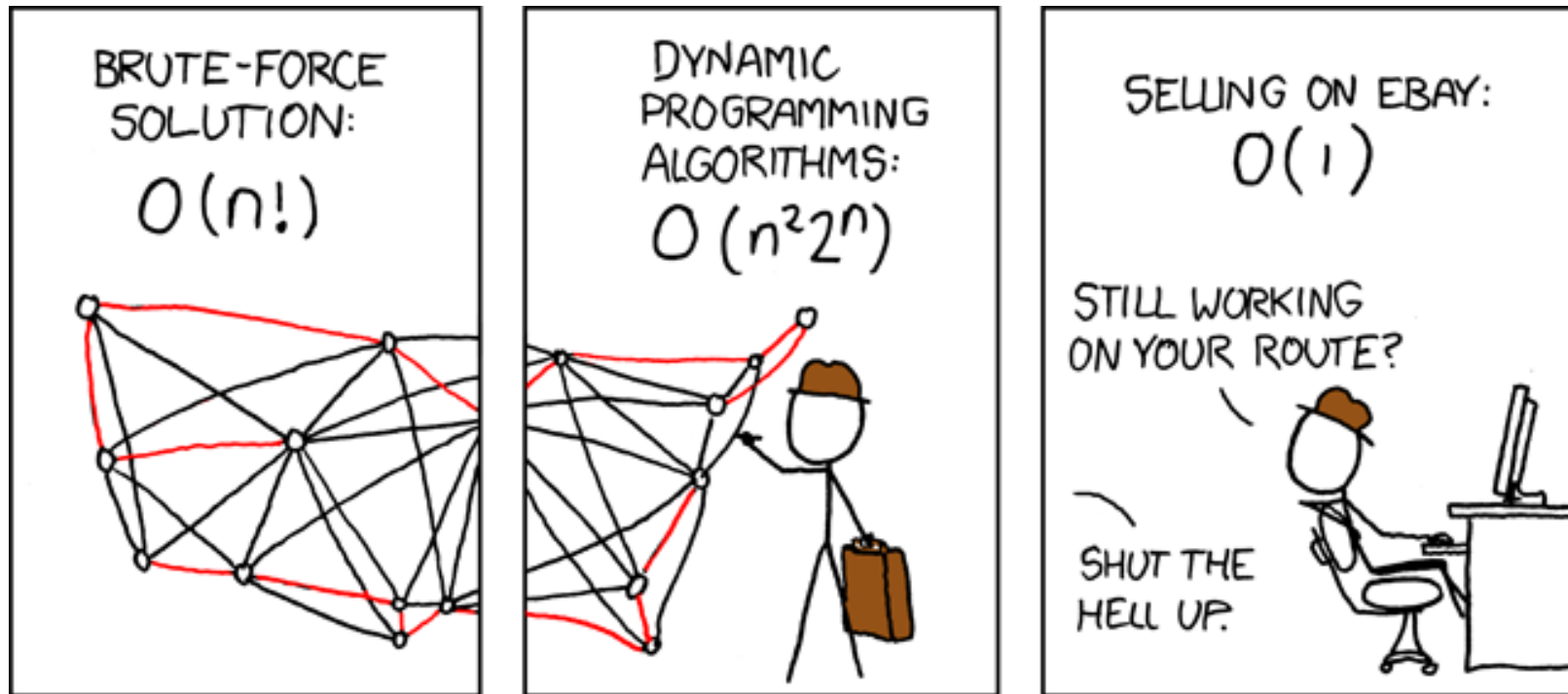


Programovacie techniky

3. „by value“, „by reference“, „reference by pointer“, quick sort, merge sort

Obchodný cestujúci

Úloha: ako navštíviť každé mesto 1 krát, tak aby bola cesta najkratšia



$O(n!)$: vyskúšaj všetky možnosti (brute force)

Zložitosť kopírovania

```
struct item {  
    int velkost;  
    int pole[velkost];  
};
```

$O(\text{velkost})$

```
struct item2 {  
    int velkost;  
    int pole[1000000];  
};
```

$O(1)$, existuje c
také, že $c \cdot 1 > 1000000$

„By value“:

```
void f(int* p) { //p je lokálna premennú, ktorú  
je možné dereferencovať
```

```
    *p = 10;  
}
```

By reference implementované ako: „reference
by pointer“

Technicky umožňuje jazyk C len „by value“

By reference:

```
void f(int& p) {  
    p = 10;  
}
```

```
int main() {  
    int a = 6;  
    f(a); //a = 10  
}
```

By reference:

```
void f() {  
    int i = 1;  
    int& r = i;  
    int x = r; //x = 1  
    r = 2;     //i = 2  
}
```

Jazyk C++

By reference a by value

Generické programovanie:

```
void push(T); //T je ľubovoľný type
```

Objektové programovanie

Merge sort

Idea: rozdelím pole na 2 časti. Tie utriedim a spojím.

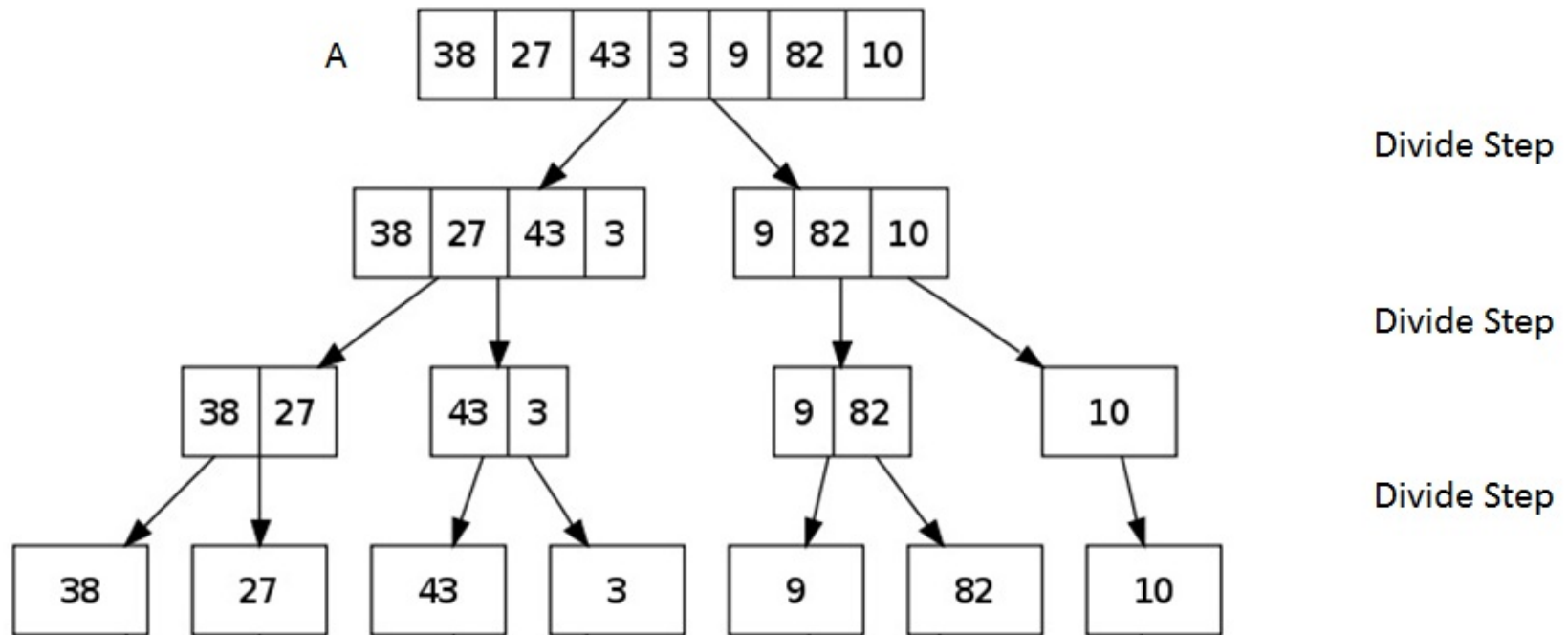
Rôzne implementácie :

1. Na jedinom statickom poli.
2. Na dvoch statických poliach – klasické riešenie
3. Na zret'azenom zozname.

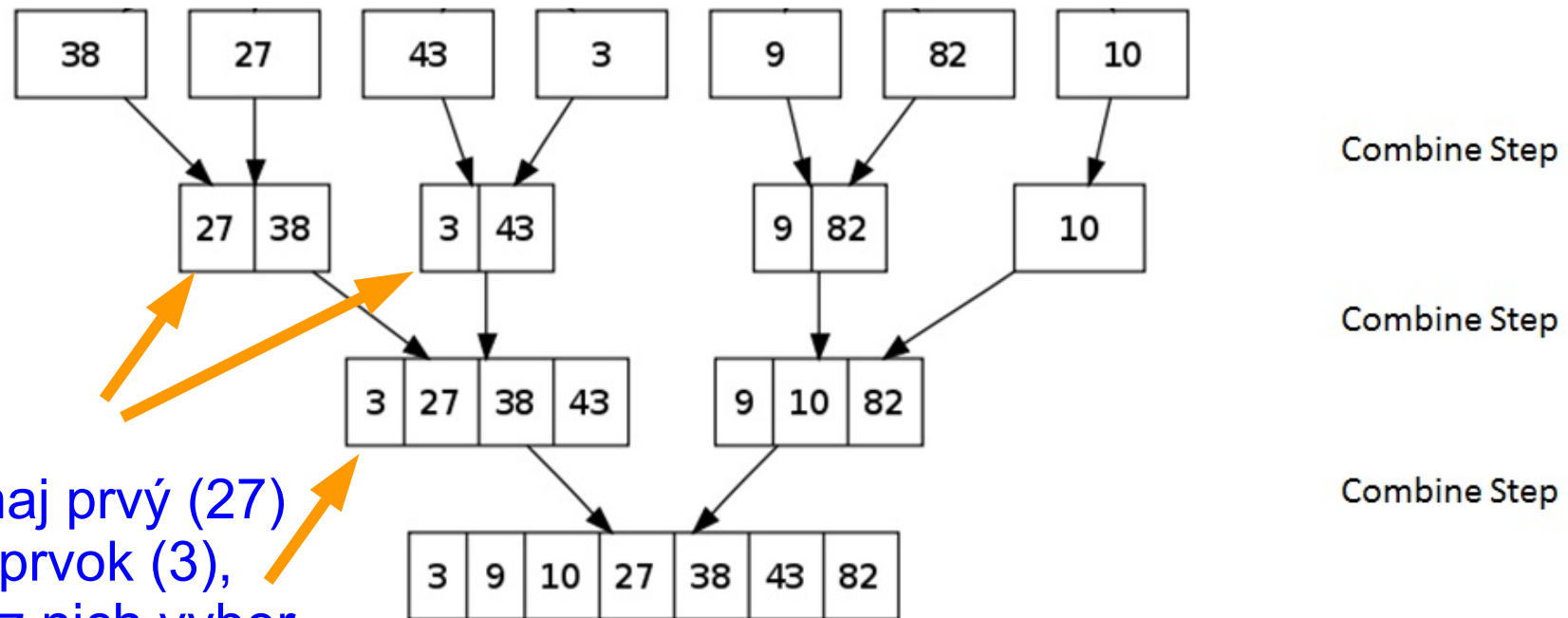
Merge sort

```
void merge_sort(int *pole, int left, int right ) {  
  
    int midle = (left+right) / 2  
    if(right == left) return;           //ukončenie vnárania  
  
    merge_sort(pole, left, middle );  
    merge_sort(pole, middle+1, right);  
  
    merge(pole, left, midle, right); //zložitost'  
}
```

Merge sort: rozdeľovanie



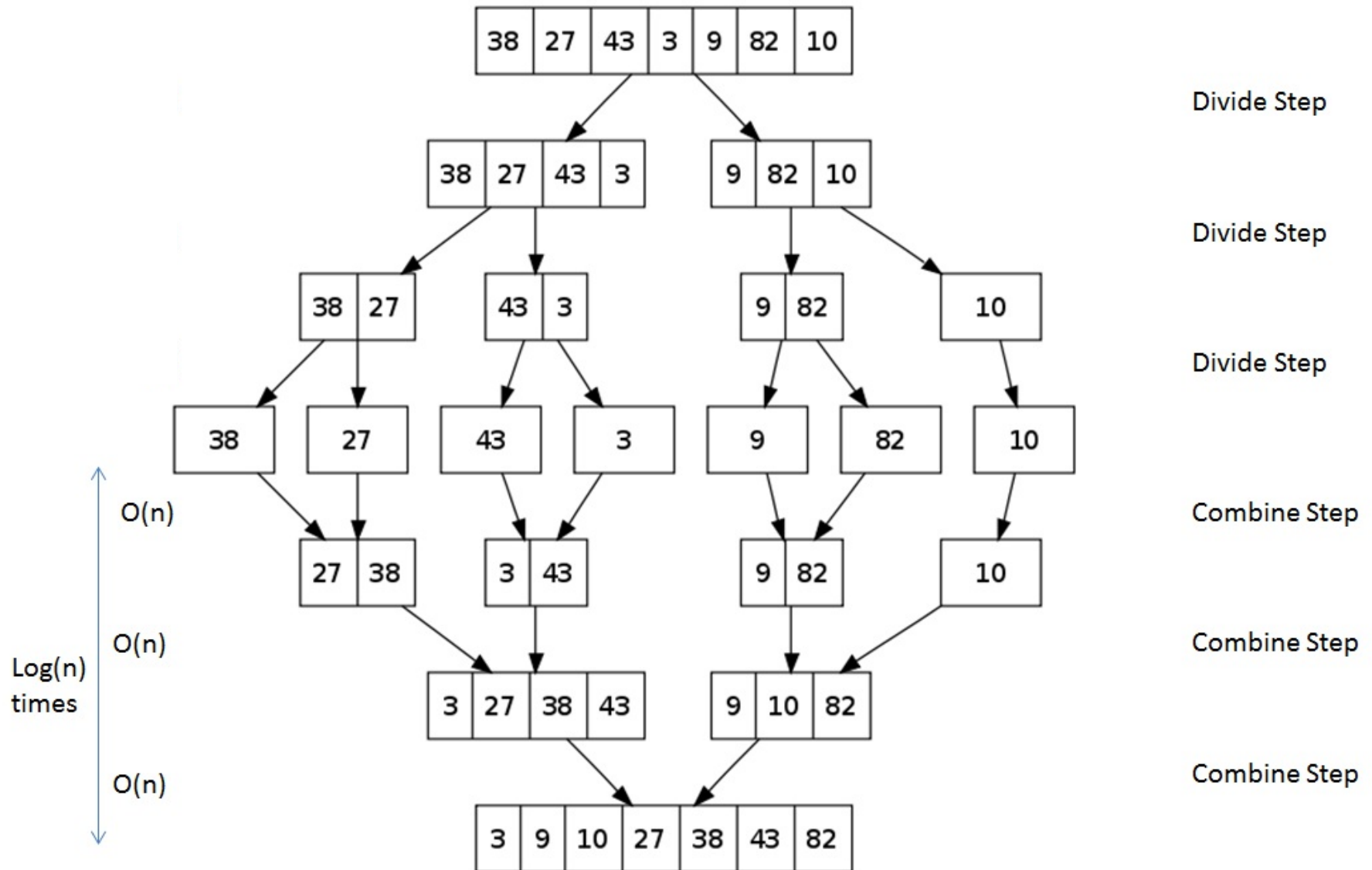
Merge sort: spájanie



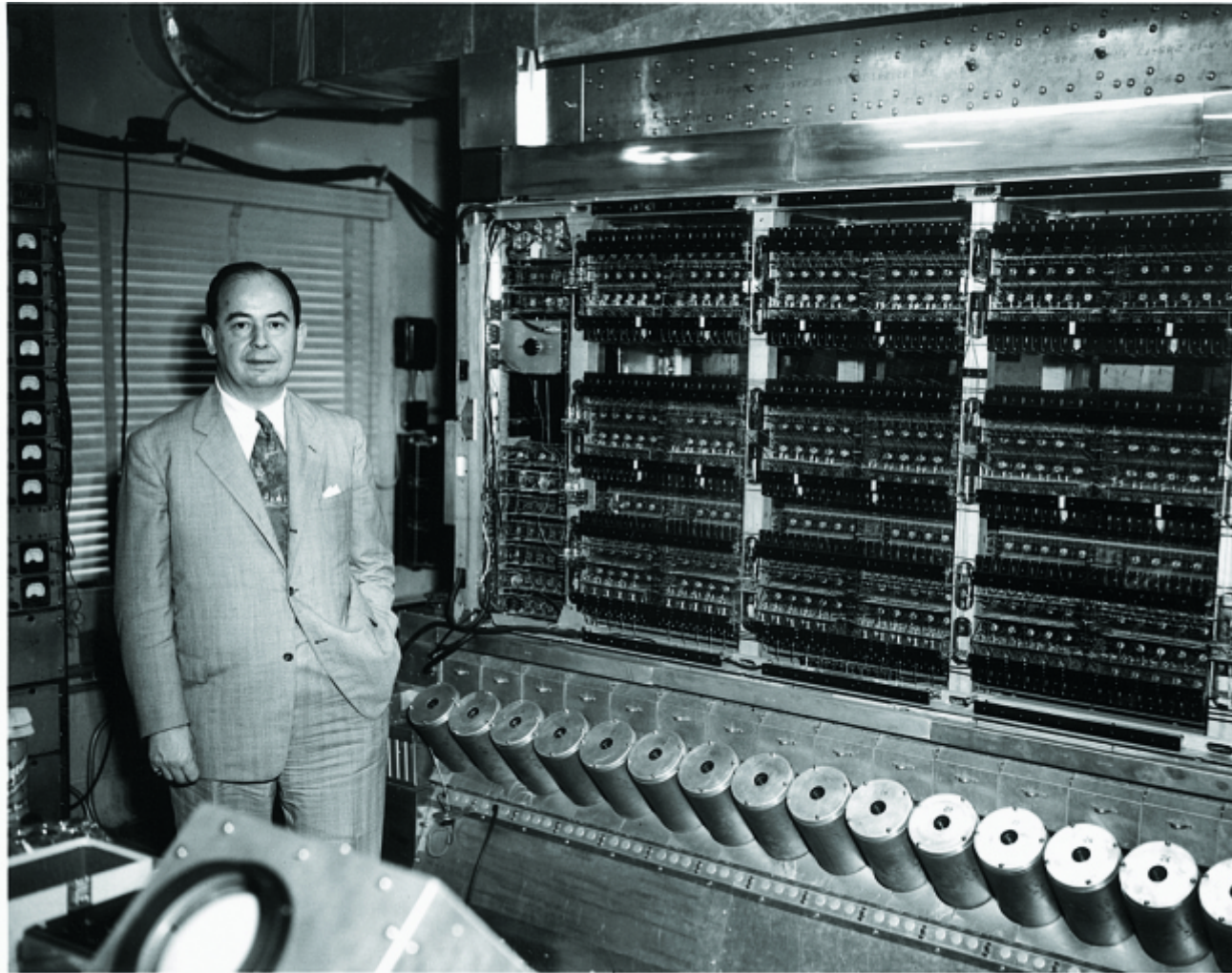
Porovnaj prvý (27)
a prvý prvok (3),
menší z nich vyber
(3) a vlož do
spojeného poľa

Je merge sort “in-place”, t.j. je dodatočné pole o
dĺžke n potrebné?

Merge sort: zložitosť



John von Neumann



Neumann János: merge sort 1945

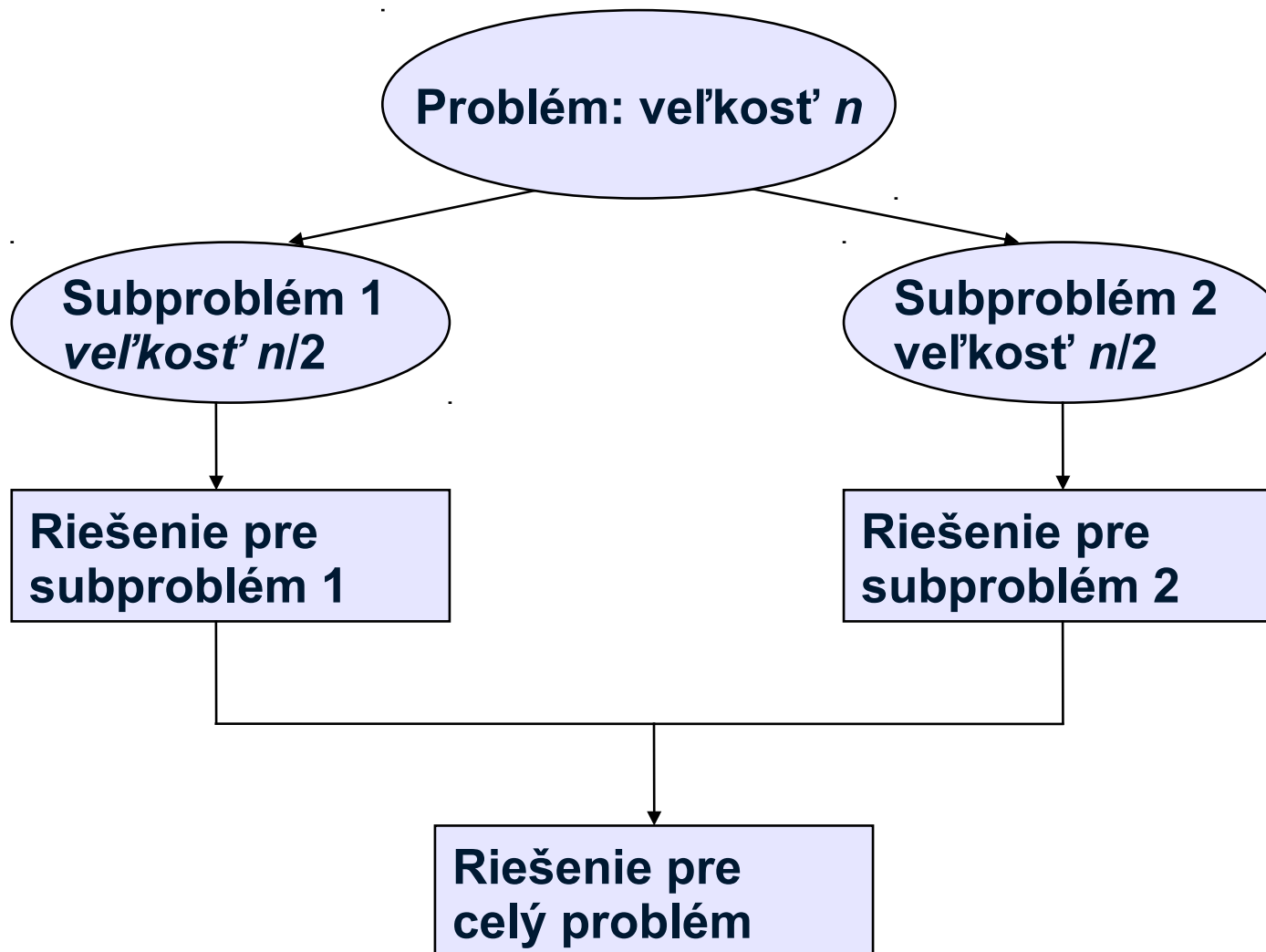
Rozdeľuj a panuj

Rozdeľ úlohu na jednoduchší problém,
nájdi riešenie, v opačnom prípade
rozdeľ úlohu ďalej.

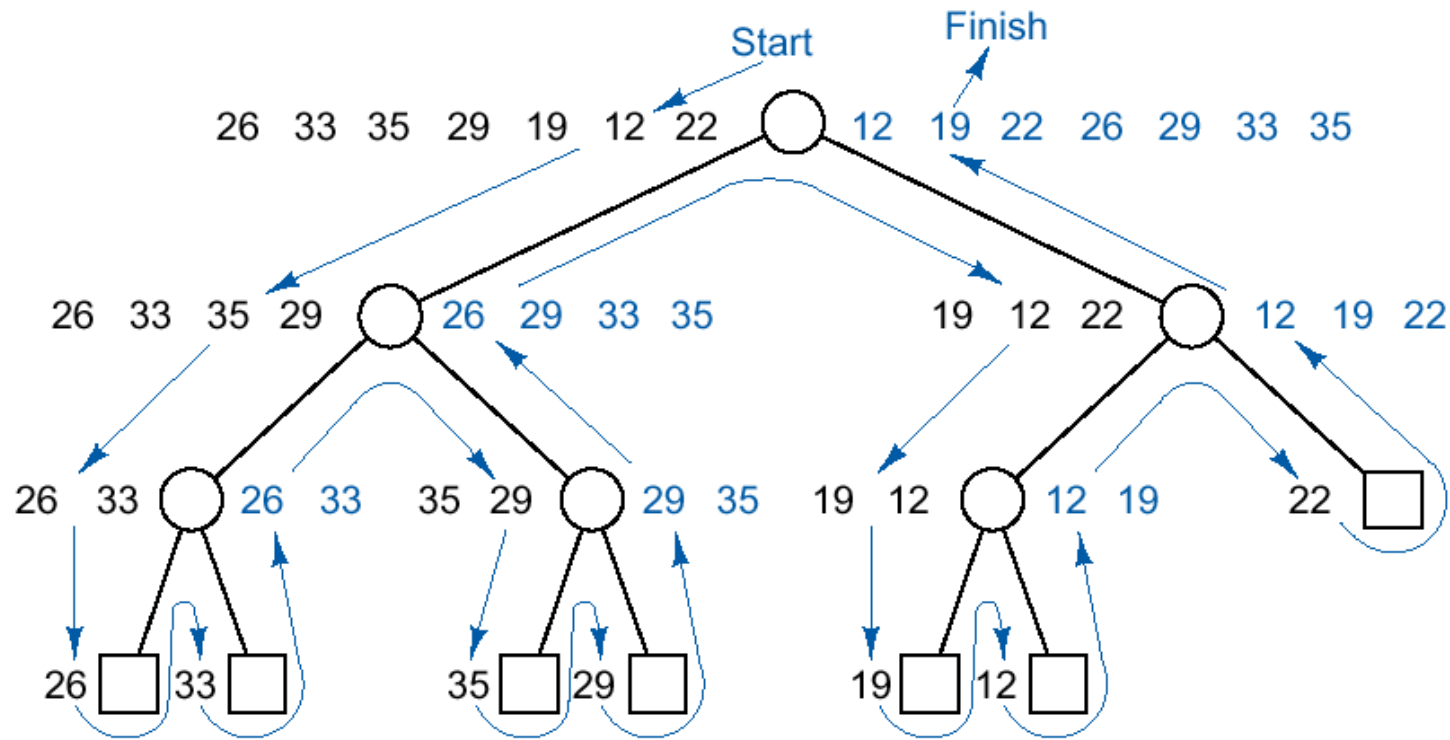
Divide and conquer

Merge sort, quick sort...
Faktoriál...

Rozdeľuj a panuj



Merge sort



Prehľadávanie do hĺbky

Quick sort

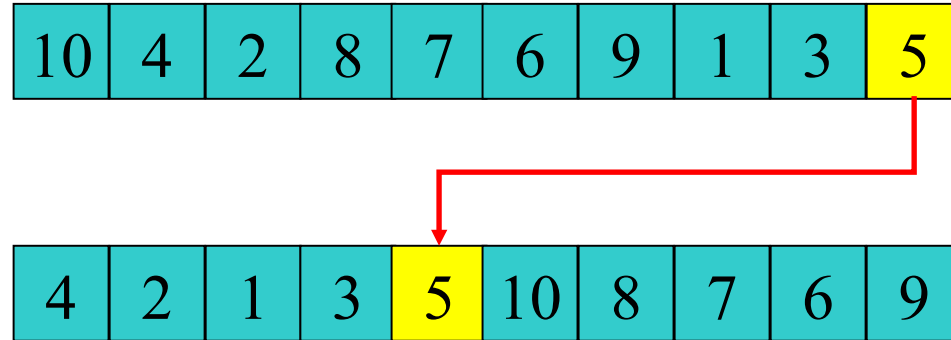
Idea:

Vyberiem z poľa prvok (pivot). Od pivota menšie prvky umiestnim naľavo a väčšie napravo.

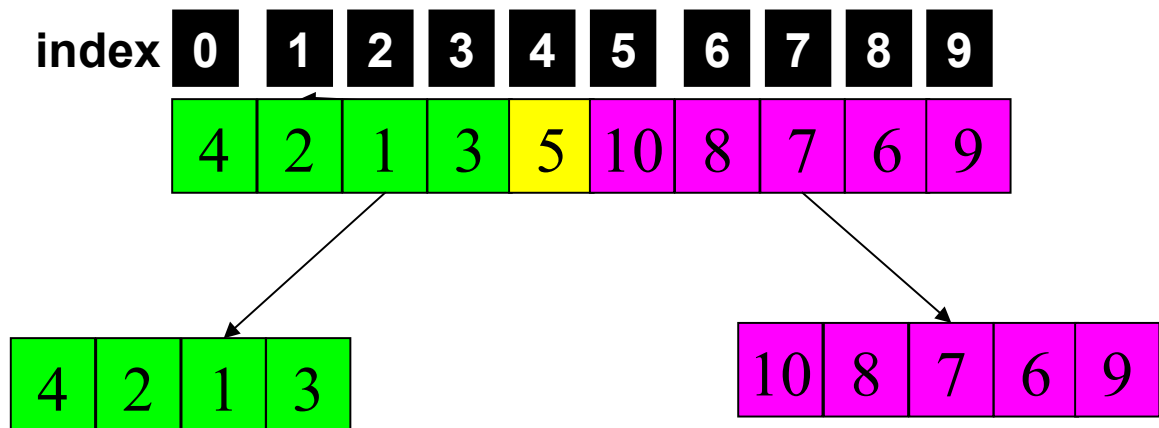
Aplikujem rekurzívne `quick_sort` zvlášť na menšie prvky a väčšie prvky.

Quick sort

1. quick_sort(0,9)



2. quick_sort(0,3)
quick_sort(5,9)

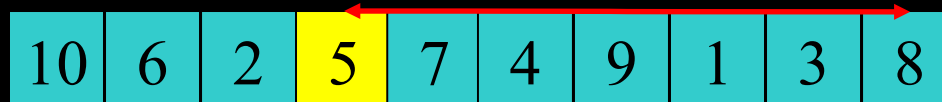


Quick sort

1. Vyberiem náhodne pivota a umiestnim ho na koniec poľa.
2. Prechod celým poľom a ukladanie za sebou na začiatok prvky menšie ako pivot.
3. Umiestnie pivota tesne za menšie prvky.

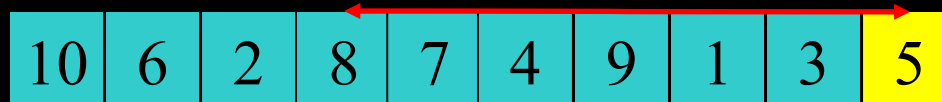
Quick sort

1. Vyberiem náhodne pivota a dám ho na koniec.



Quick sort

1. Vyberiem náhodne pivota a dám ho na koniec.

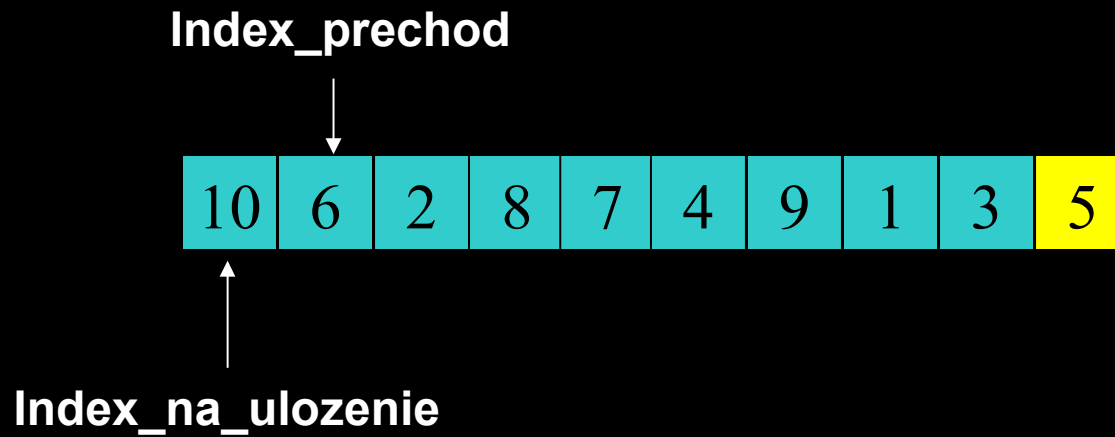


Quick sort

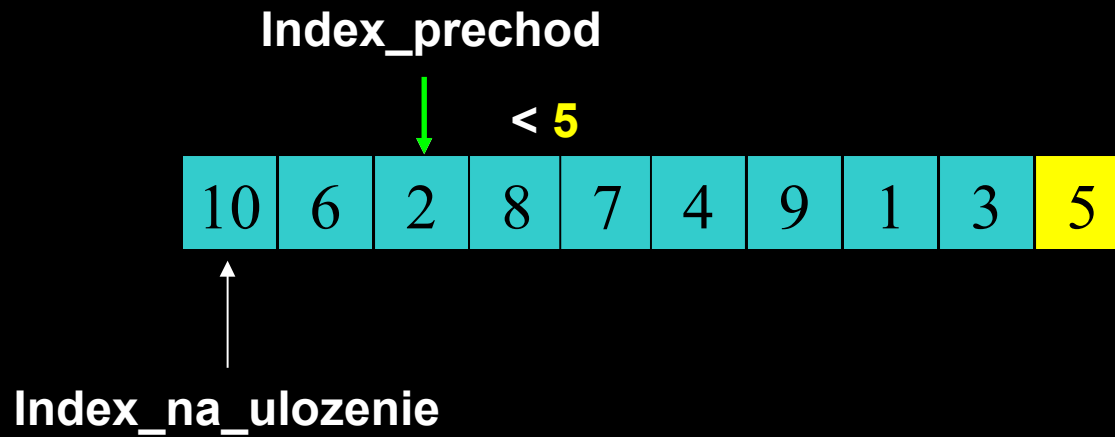
2. Prechod celým poľom a ukladanie za sebou na začiatok prvky menšie ako pivot.



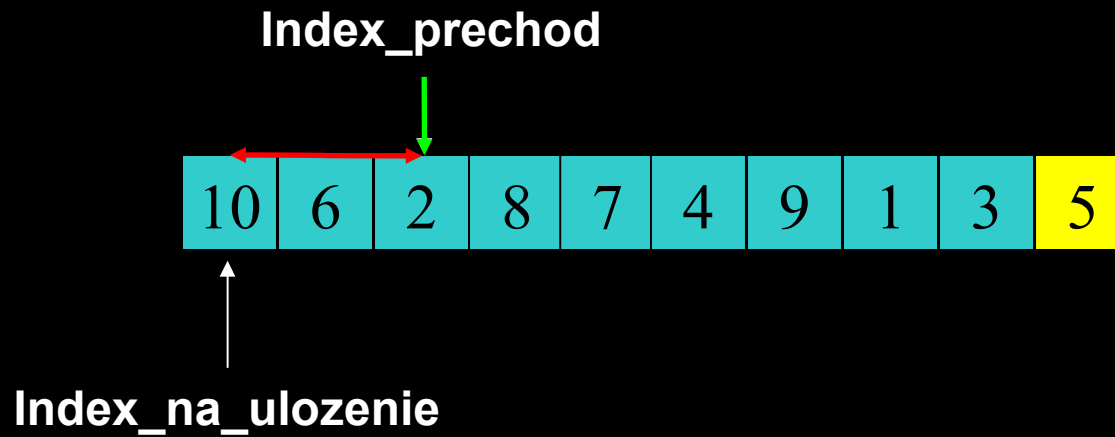
Quick sort



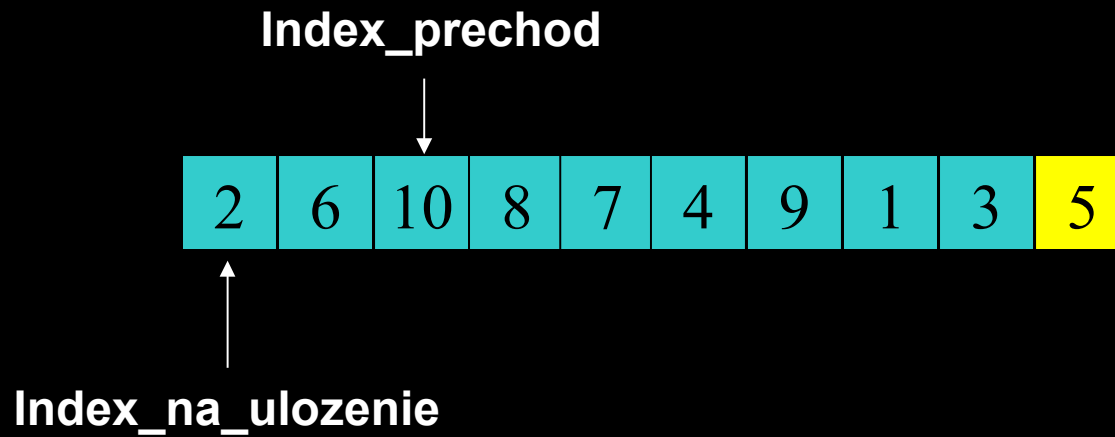
Quick sort



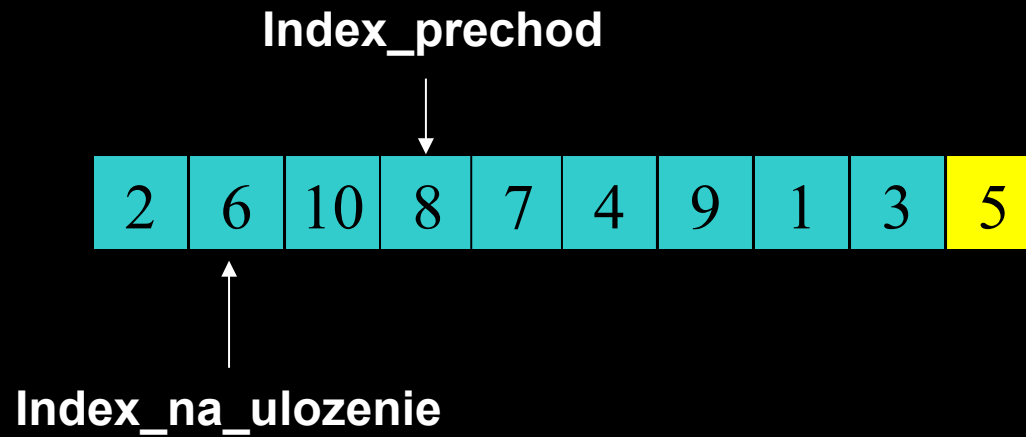
Quick sort



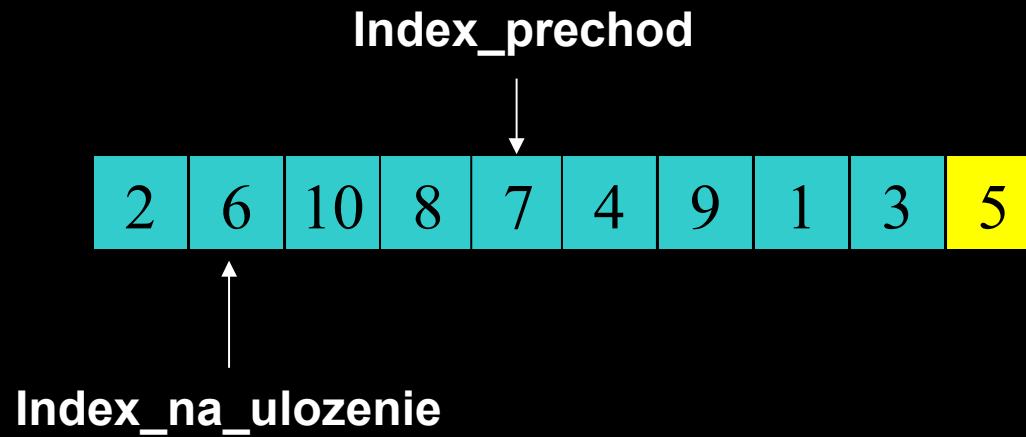
Quick sort



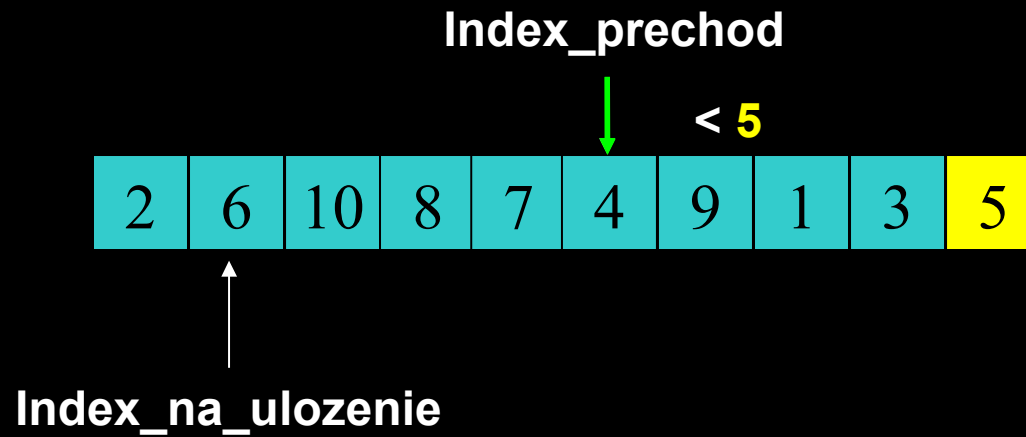
Quick sort



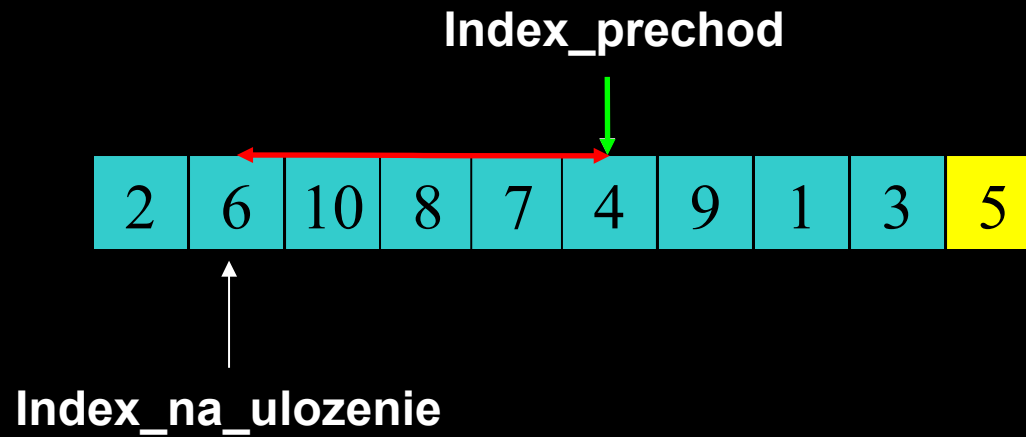
Quick sort



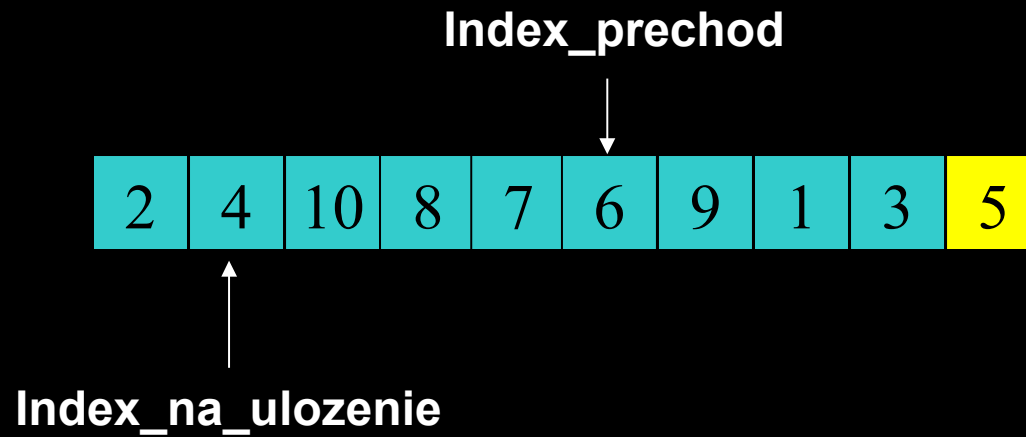
Quick sort



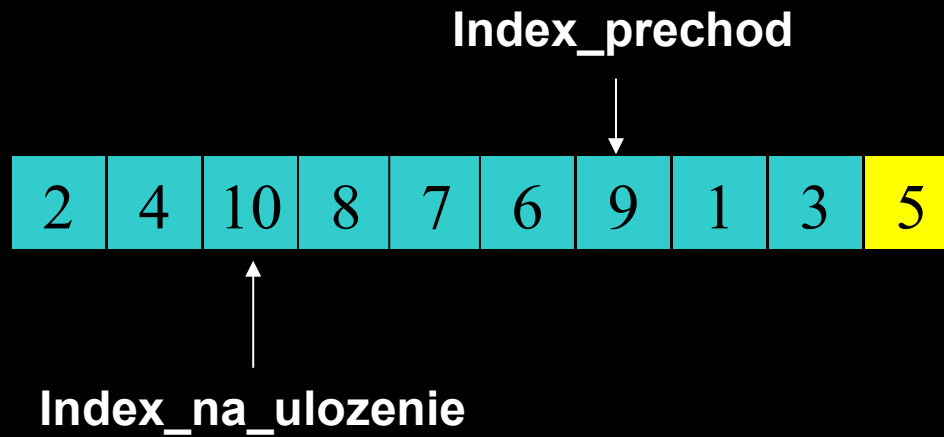
Quick sort



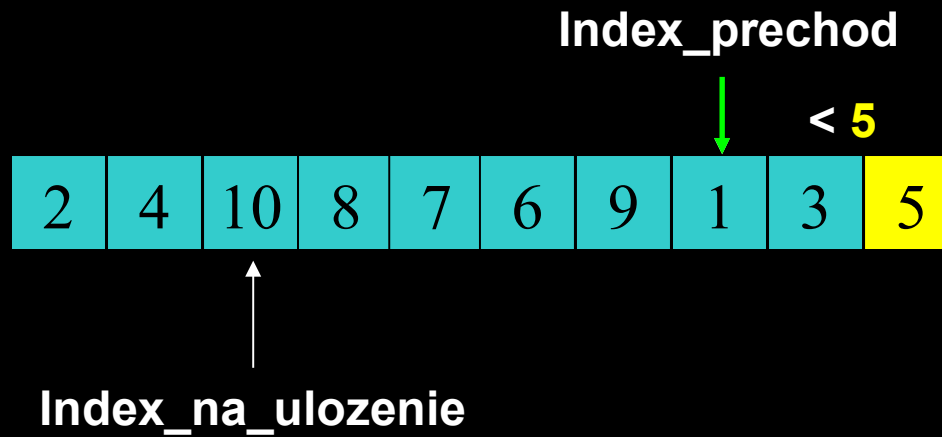
Quick sort



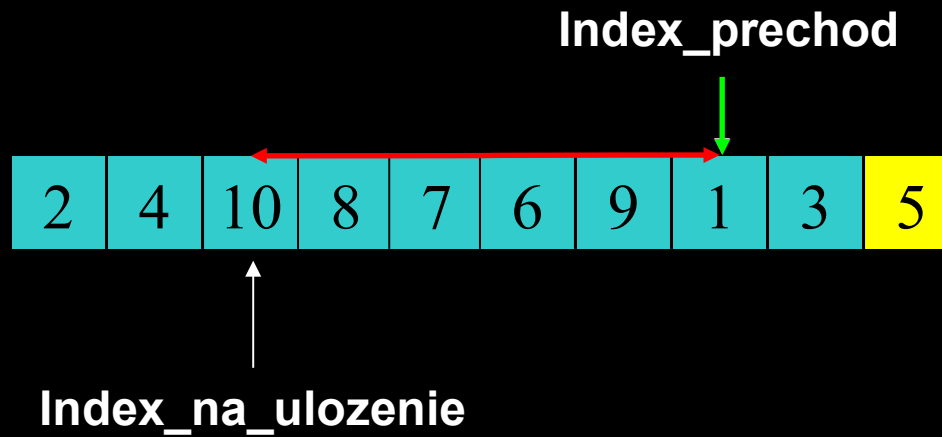
Quick sort



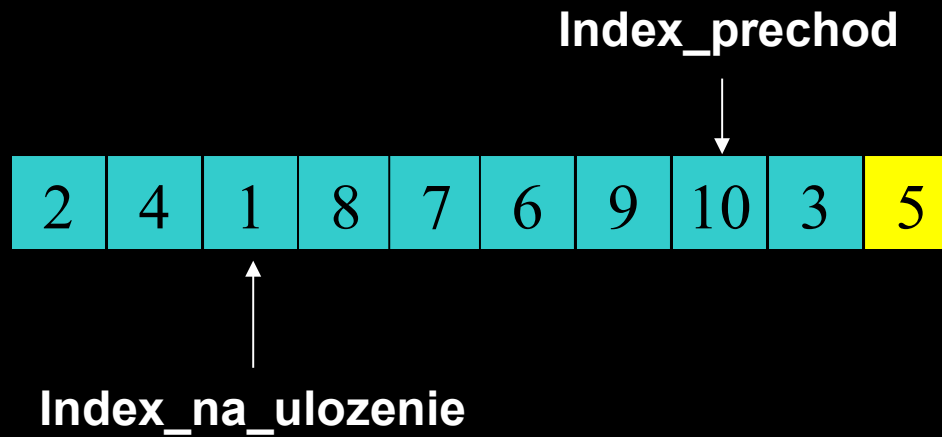
Quick sort



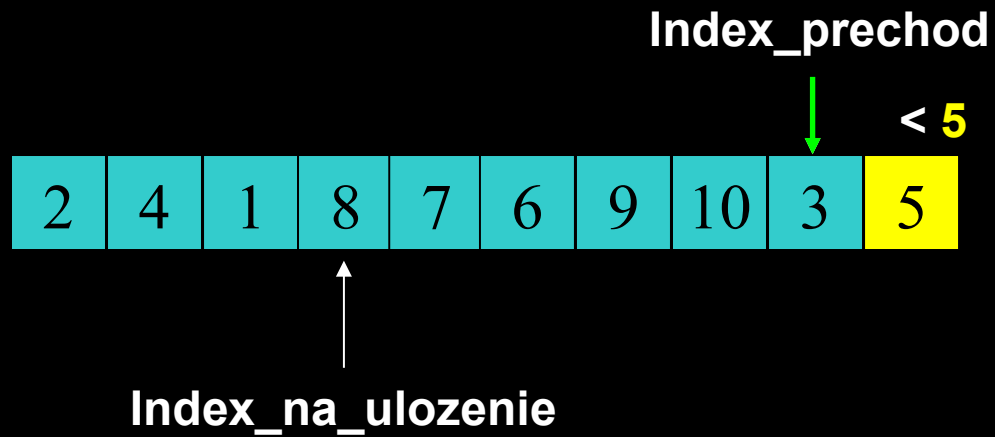
Quick sort



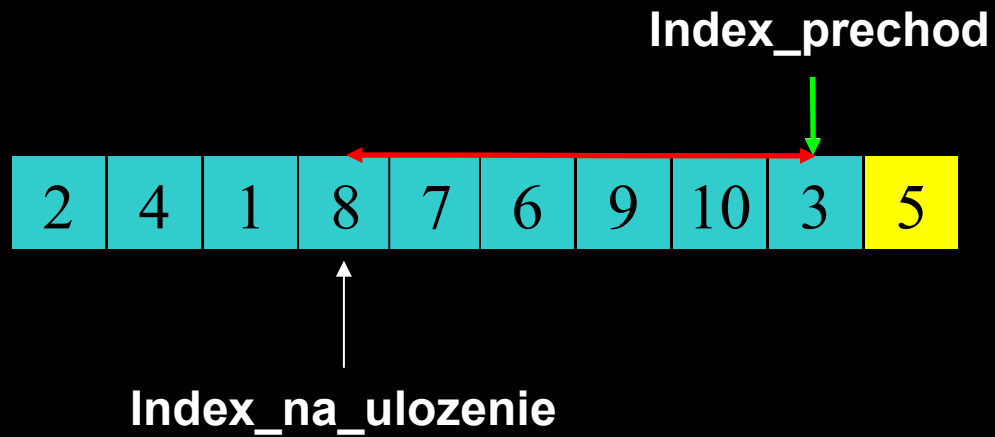
Quick sort



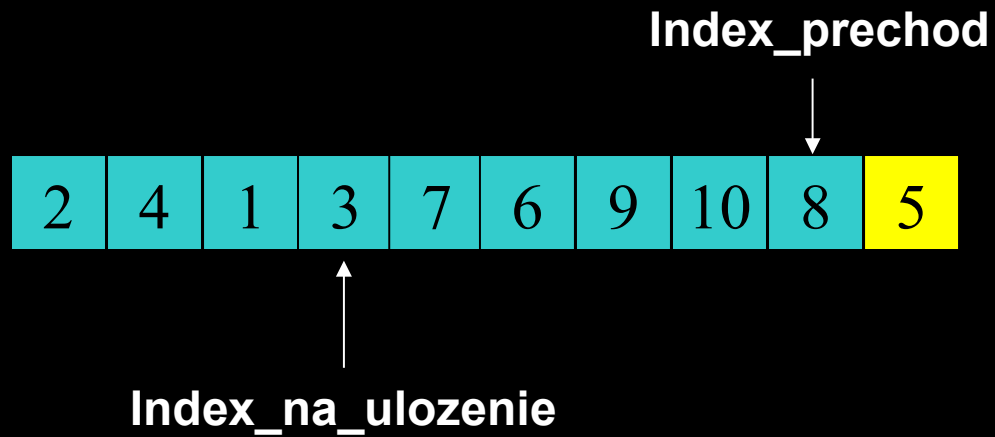
Quick sort



Quick sort



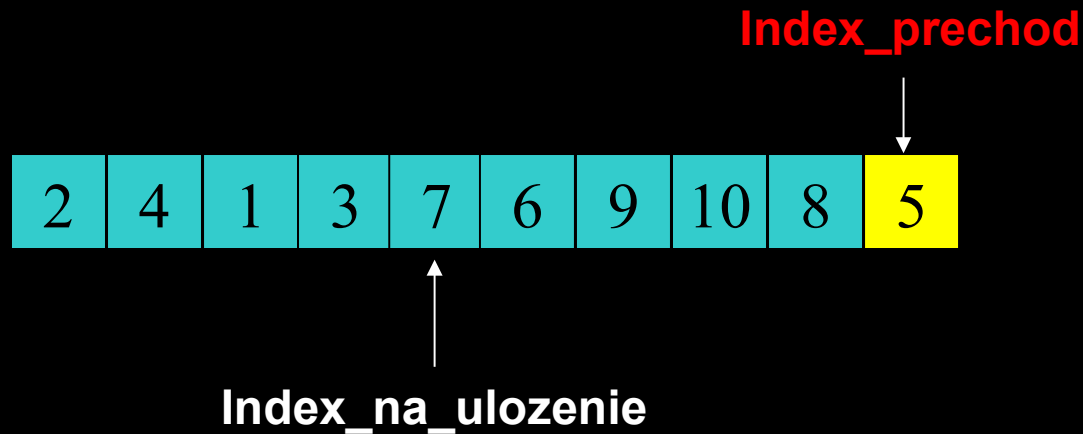
Quick sort



Quick sort

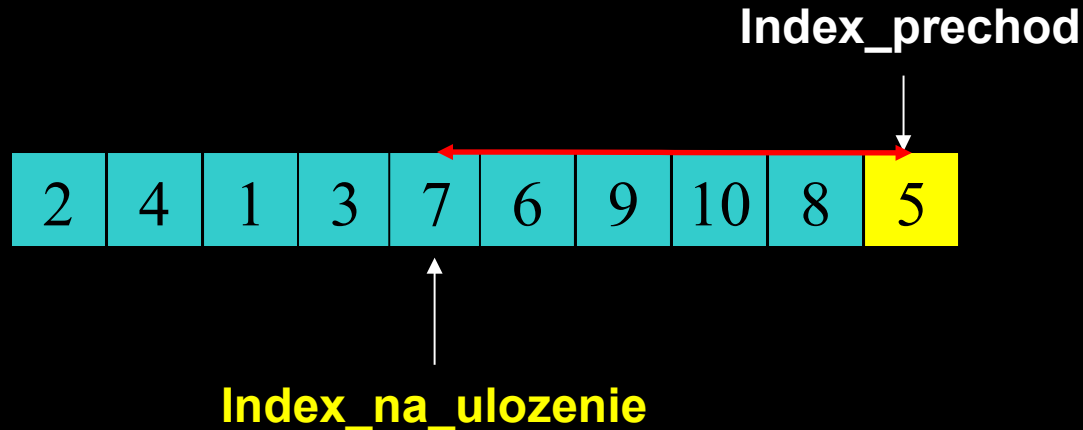
Koniec prechodu:

index_prechod = **index pivota**



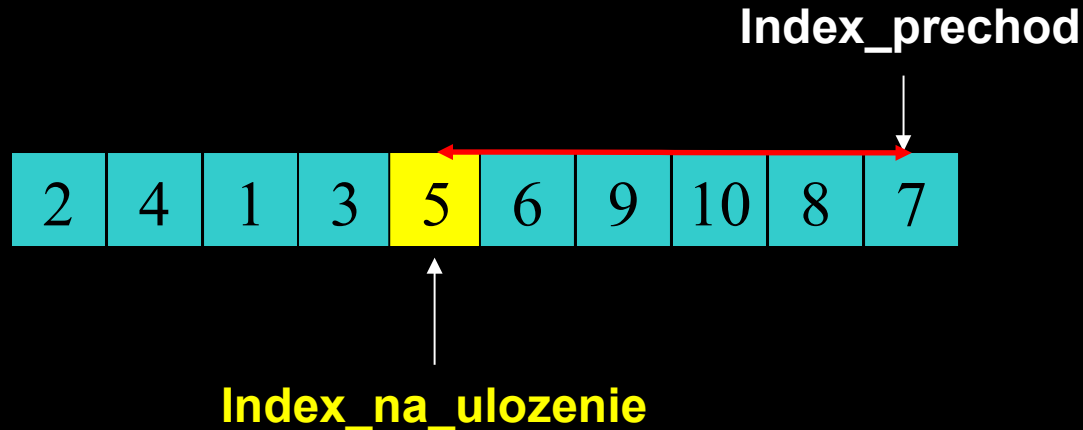
Quick sort

3. Umiestnim pivota na správne miesto
index_na_umiestnenie



Quick sort

3. Umiestnim pivota na správne miesto
index_na_umiestnenie



Voľba pivota

Prvok vľavo/vpravo: zlá voľba, ak je pole už usporiadané

Náhodná voľba

Medián: prvý, posledný a stredný prvok

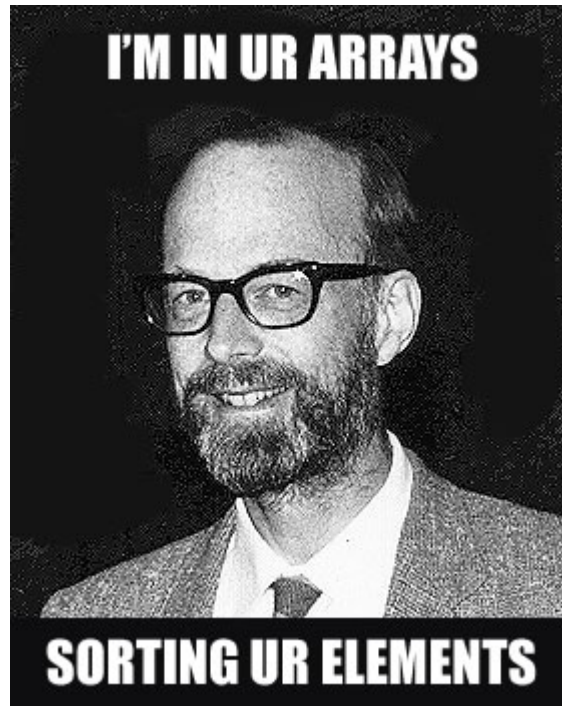
Quick sort: zložitosť

Najhorší prípad: pivot je najväčší prvok v každom kroku, dve časti o veľkosti $n-1$ a 0 : $O(n^2)$

Najlepší prípad: rozdelenie v každom kroku na dve rovnaké časti: $O(n \log n)$, hĺbka stromu volaní $\log n$

Priemerný prípad: lepšie a horšie rozdelenia sa kompenzujú: $O(n \log n)$, hĺbka stromu volaní $O(\log n)$

Tony Hoare



Quick sort: 1960 počas študijného pobytu v ZSSR

Otázka za milión

Aký je najlepší triediaci algoritmus, ktorý existuje?

To, že vieme triediť v $O(n \log n)$ neznamená, že neexistuje rýchlejší algoritmus.