Pokročilé Dátové Štruktúry

Pokročilé dátové štruktúry a čo ukladajú

Dátová Štruktúra	Čo Ukladá	Poznámky
array	ukladá prvky v konkrétnom poradí	v Pythone to isté ako listy
queue	ukladá prvky v konkrétnom poradí	prvý dnu, prvý von (FIFO)
stack	ukladá prvky v konkrétnom poradí	posledný dnu, prvý von (LIFO)
tree	ukladá prvky hierarchicky	každý uzol môže mať viacero detí
binary tree	ukladá prvky hierarchicky	každý uzol má maximálne 2 deti
heap	ukladá prvky podľa priority	vždy dostaneš najmenší/najväčší prvý

Algoritmy na Triedenie

Časová komplexita (Big O)

Čo to je? Metrika, ktorá opisuje, ako sa čas behu zväčšuje s veľkosťou inputu
Prečo to potrebujeme? Na porovnávanie efektivity algoritmov na veľkých dátach
Ako to vypočítam? Spočítaj operácie: konštanty = O(1), loops = O(n), nested loops = $O(n^2)$

Príklady Algoritmov na Triedenie

Bubble Sort (pomalý)

```
function bubble_sort(array):
    n = length of array

    for i from 0 to n-1:
        for j from 0 to n-2:
            if array[j] > array[j+1]:
                swap array[j] and array[j+1]

    return array
```

Merge Sort (rýchly)

```
function merge_sort(array):
    if length of array <= 1:
        return array

    mid = length of array / 2
    left = merge_sort(array[0 to mid])
    right = merge_sort(array[mid to end])

    return merge(left, right)
function merge(left, right):
    result = empty array
    i = 0, j = 0

    while i < length of left AND j < length of right:
        if left[i] < right[j]:
            append left[i] to result
            i = i + 1
        else:
            append right[j] to result
            j = j + 1

    append remaining elements from left to result
    append remaining elements from right to result

    return result
```