

# Złożoność algorytmów sekwencyjnych.

$D_n$  - zbiór danych wejściowych rozmiaru  $n$

$t(d)$  - liczba **operacji dominujących** wykonywanych przez algorytm po uruchomieniu dla danych wejściowych  $d \in D_n$

$X_n$  - zmienna losowa, której wartością jest  $t(d)$  dla  $d \in D_n$

**def:** pesymistyczna złożoność czasowa algorytmu (ang. worstcase complexity) to  $W(n) = \sup\{t(d) : d \in D_n\}$

**def:** pesymistyczna wrażliwość algorytmu to  $\Delta(n) = \sup\{t(d_1) - t(d_2) : d_1, d_2 \in D_n\}$

**def:** oczekiwana złożoność czasowa algorytmu to  $A(n) = E(X_n)$

**def:** oczekiwana wrażliwość algorytmu to  $\delta(n) = \sqrt{\text{Var}(X_n)}$  czyli odchylenie standardowe  $X_n$

**def:** złożoność pamięciowa - ile pamięci wymaga algorytm

Algorytmy o złożoności czasowej wielomianowej, np.:

- + sortowanie,  $O(n^2)$ ,  $O(n \log n)$ , ...
- + wyszukiwanie proste i binarne w tablicy n-elem
- + scalanie macierzy z posort wierszami
- + co jeśli rozmiar danych we jest opisany 2 parametrami ???

Klasa problemów wielomianowych (P)

takich dla których istnieje algorytm wielomianowy

NP, NP-zupełność... (o tym później)

*czy warto szukać alg wielomianowego ??*

Złożoność pamięciowa:

- + MergeSort ( $2n$ ), QuickSort ( $n$ )

Wrażliwość algorytmu:

- + porównaj sortowanie **przez selekcję** i **przez wstawianie**  
zbadaj posort i nie posort dane wejściowe...

Zbadaj złożoność czasową i pamięciową algorytmów:

**lista prostych problemów dla tablic**

# Proste struktury danych: kolejki, stosy, listy

patrz **opis stosów i kolejek**

Na każdej strukturze wykonuje się operacje:

Pisz, Czytaj, Inicjuj

(Cormen: Push, Pop, Enqueue, Dequeue)

Kolejka ma skończony bufor,  
indeksy p i c się „przewijają” !

patrz **opis list**

Listy jedno i dwu-kierunkowe, z wartownikiem i bez

Jak bezpiecznie programować operacje na listach ??

Zmienne dynamiczne (vs „statyczne”):

- + tworzone i niszczone „na żądanie” (jak to jest w Pytonie?)
- + wskaźnik/ referencja
- + sarta (obszar dla tych zmiennych),
- + automatyczne usuwanie zm. dyn. czyli „odśmiecanie”  
(ang. garbage collection) gdy nie ma już wsk/ref do zm...

# Drzewiaste struktury danych...

Patrz: slajdy wykł10\_MH\_bst.pdf  
alt opisy drzew **BST**, **Z.22** i drzew **RB**

BST = Binary Search Tree

Motywacja drzew BST: *szybki dynamiczny słownik*

pary (klucz, wartość)

operacje na słowniku:

insert(S, elem), delete(S,elem), elem=search(S,klucz)

słownik „statyczny”:

najpierw tylko insert() potem tylko search

słownik „dynamiczny”:

pomieszane operacje insert/search/delete

czy słownik dynamiczny można zrobić na tablicy ?

(czy to będzie szybki słownik ??)

