

LABORATORIUM NR 1

---

EKSPERYMENTALNA ANALIZA CZASU DZIAŁANIA

---

**Zadanie ALL.1.1**

Poniżej podane są przykładowe pseudokody trzech procedur oraz implementacja pierwszej z nich, wraz z przykładowym sposobem pomiaru czasu działania (patrz także plik ALL\_01.c). Przeanalizuj pseudokody tych procedur i „pobieżnie” oszacuj, jaka jest złożoność czasowa każdej z tych procedur. W oparciu o przykładowy pomiar czasu działania przetestuj doświadczalnie oszacowanie dla pierwszej z nich. Następnie zaimplementuj pozostałe procedury i również dokonaj doświadczalnego pomiaru czasu, porównując go z zaproponowanym oszacowaniem.

- Procedura1(n:integer)  
begin  
x:=0.0;  
for i:=n downto 1 do  
if nieparzyste(i) then begin  
for j:=1 to i do ;  
for k:=i+1 to n do x:=x+1;  
end  
return x;  
end
- Procedura2(A[1,...,n])  
begin  
x:=0.0;  
for d:=1 to n do  
for g:=d to n do begin  
suma:=0.0;  
for i:=d to g do  
suma:=suma+A[i];  
x:=max(x,suma);  
end  
return x;  
end

Co oblicza powyższa Procedura2?

- Procedura3(n:integer)  
begin  
for i:=1 to sqrt(n) do begin  
j:=1;  
while j<sqrt(n) do j:=j+j;  
end  
end

### Przykładowy pomiar czasu (w pliku ALL\_01.c).

/© P. Pączkowski/

- ▶ Wywołujemy badaną procedurę dla różnych wartości  $n$ , np.  $n = 5, 10, 15, \dots$ , mierząc w programie rzeczywisty czas  $T(n)$  działania odpowiedniej procedury /zmienna  $Tn$ /.
- ▶ Następnie, jeśli  $F(n)$  jest oszacowaniem na czas procedury /zmienna  $Fn$ /, np.  $F(n) = 5 \cdot n$ , wyliczamy ilorazy  $F(n)/T(n)$ .
- ▶ Jeżeli czas rzeczywisty zgadza się z teoretycznym oszacowaniem, wówczas otrzymane ilorazy powinny wyjść mniej więcej stałe.

```
// kompilowac z opcjami -lrt -lm, tj. np. gcc ALL_01.c -lrt -lm
#include <time.h>
#include <stdio.h>
#include <math.h>
#define MLDD 1000000000.0 //10**10

double procedural1(int n){
    float x=0;
    int i,j,k;

    for(i=n;i>=1;i--) {
        if((i % 2) == 1) {
            for(j=1;j<i+1;j++) ;
            for(k=i+1;k<n+1;k++) x=x+1;
        }
    }
    return x;
}

int main(){
    struct timespec tp0, tp1;
    double Tn,Fn,x;
    int n;

    for(n=10;n<10011;n=n+100){
        clock_gettime(CLOCK_PROCESS_CPUTIME_ID,&tp0);
        // przykladowe obliczenia
        x=procedural1(n);
        clock_gettime(CLOCK_PROCESS_CPUTIME_ID,&tp1);
        // zgadywana funkcja czasu
        Fn=5*n ; // np. funkcja liniowa
        //     Fn=2000*n;
        //     Fn=n*n*n;
        //     Fn=n*log(n);
        //     Fn=n*n*sqrt(n);
        //     Fn=n*n;
        Tn=(tp1.tv_sec+tp1.tv_nsec/MLDD)-(tp0.tv_sec+tp0.tv_nsec/MLDD);
        printf("n: %5d \tczas: %3.10lf \twspolczynnik: %3.5lf\n",n,Tn, Fn/Tn);
    }
    return 1;
}
```