

Algorytmy i struktury danych I

Poprawność semantyczna

- Zadanie 1 (tablicowe:omawiamy w trakcie ćwiczeń)
Sprawdź poprawność specyfikacji:

```
/* x >= 0 */ (warunek wstępny)
x = x+1
/* x >= 1 */ (warunek końcowy)
```

- Zadanie 2 (tablicowe:omawiamy w trakcie ćwiczeń)
Sprawdź poprawność specyfikacji dla sekwencyjnego wykonania instrukcji

```
/* x >= 1 */ (warunek wstępny)
y = x
/* y >= 1 */ (warunek końcowy)
```

```
/* y >= 1 */ (warunek wstępny)
y = y+1
/* y >= 2 */ (warunek końcowy)
```

- Zadanie 3 (tablicowe:omawiamy w trakcie ćwiczeń)
Udowodnij poprawność warunku końcowego dla następującego kodu.

```
/**/ (warunek wstępny)
if( x>=0 ) y=y+x
else y = y-x
/* y = y0 + |x0| */(warunek końcowy)
```

- Zadanie 4
Podaj niezmiennik dla poniższej pętli i udowodnij jego poprawność:

```
/* 1 <= b */ (warunek wstępny)
i=1; c=1;
while ( i<= b) { c = c * a ; i++};
```

- Zadanie 5 (tablicowe:omawiamy w trakcie ćwiczeń)
Udowodnij poprawność warunku końcowego dla następującego kodu:

```
/*x >= 100 */
if( x >= y )
    m = x;
else
    m = y;
/* m >= 100 */
```

- Zadanie 6
Napisz krótki program, który wczytuje ciąg liczb dodatnich, zakończony przez -1, oblicza i wypisuje największą z wczytanych liczb dodatnich. Jaki jest niezmiennik dla pętli w Twoim programie? wypisuj jego wartość.
- Zadanie 7
Napisz krótki program który wczytuje ciąg 100 liczb dodatnich, oblicza i wypisuje największą z wczytanych liczb dodatnich. Jaki jest niezmiennik dla pętli w Twoim programie? wypisuj jego wartość.
- Zadanie 8
Napisz krótkie programy realizujące następujące zadania, postaraj się znaleźć algorytm o złożoności lepszej od złożoności kwadratowej.
 - **Szukanie przywódcy ciągu:** Przywódcą ciągu jest element, który występuje w ciągu więcej razy niż połowa długości tego ciągu. Naszym problemem jest policzenie przywódcy ciągu w tablicy $A[1..n]$. Dla uproszczenia przyjmijmy, że w tym ciągu jest przywódca. Łatwo zmodyfikować algorytm tak, by sprawdzał istnienie przywódcy.
 - **Szukanie sumy:** Mamy dane dwie tablice posortowane rosnąco A, B i liczbę x , pytamy, czy istnieją $a \in A$, $b \in B$ takie, że $x=a+b$.
 - **Szukanie maksymalnego segmentu:** Dla tablicy $A[1..n]$ liczymy maksymalną wartość z zera i ze wszystkich liczb $\sum_{k=i}^j A[k]$, gdzie $1 \leq i \leq j \leq n$.
 - **Szukanie najdłuższego malejącego podciagu:** Niech $A[1], A[2] \dots A[n]$ będzie ciągiem n dodatnich liczb. Oblicz długość najdłuższego malejącego podciagu (w kolejności od lewej do prawej strony).

Złożoność obliczeniowa

Zadania tekstowe przygotowujemy w postaci dokumentu: plik.pdf (używamy latex, word albo scan z ręcznego pisma) i umieszczamy w systemie Pegaz.

- Zadanie 1 (tekstowe)

Uporządkuj podane niżej funkcje wg. asymptotycznego stopnia złożoności tak, aby każda funkcja była asymptotycznie mniejsza od następujących po niej:

$$51n+101, \frac{n^3}{7lg^7n}, \frac{n^2+2}{lgn}, (\sqrt{n} + 1)^3, \frac{lgn}{n}, \frac{n}{lgn}, \sum_{k=0}^n k\sqrt{k}.$$

- Zadanie 2 (tekstowe)

Korzystając z twierdzenia o rekursji uniwersalnej oszacuj rząd wielkości funkcji T zadanej równaniem rekurencyjnym:

- $T(n) = 4T(\frac{n}{2}) + n$

- $T(n) = 4T(\frac{n}{2}) + n^2$

- $T(n) = 4T(\frac{n}{2}) + n^2lg_2n$

- $T(n) = 4T(\frac{n}{2}) + n^3$

- $T(n) = T(\frac{n}{2}) + c$

- Zadanie 3

Wykorzystaj program z zestawu 2 (implementacja szybkiego algorytmu sortującego). Zliczaj ilość wykonywanych porównań. Wpisz tabelkę lub narysuj wykres zależności iloci porównań od iloci elementów N sortowanego ciągu liczb. Jak wygląda równanie rekurencyjne dla zaimplementowanego algorytmu. Czy ilość porównań zachowuje się tak jak przewiduje rozwiązanie równania rekurencyjnego dla tego algorytmu?