

SUMA ITERACYJNIE

1. Dane zadane:

$$\{a_1, a_2, a_3, \dots, i, \dots, n\} = \{1, 2, 3, \dots, i, \dots, n\} \quad n \in \mathbb{N}$$

2. Onehildany mymih:

$$S(n) = \sum_{i=1}^n i$$

3. Analiza rekurencja

$$S(1) = a_1$$

$$S(2) = a_1 + a_2$$

$$S(3) = S(2) + a_3$$

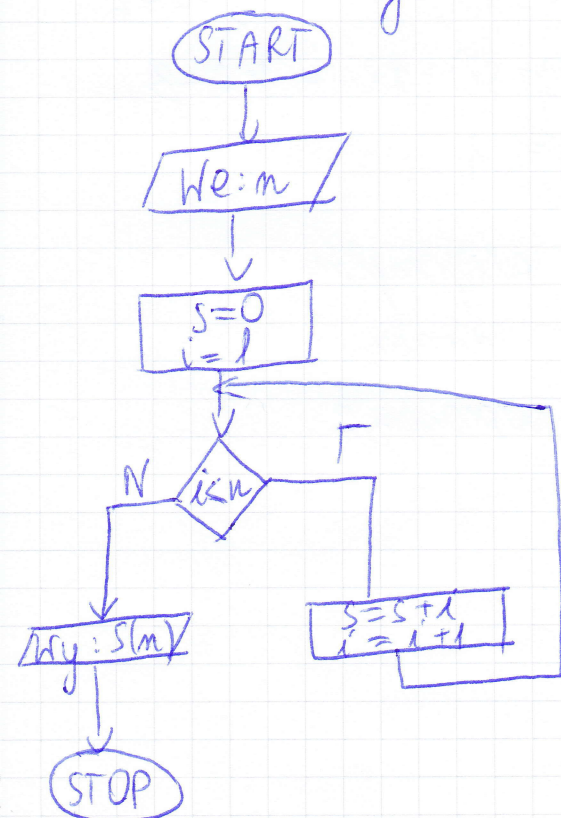
$$S(i) = S(i-1) + i$$

$$S(n) = S(n-1) + n$$

$$S(i) = \begin{cases} S(i-1) + i & \text{dla } i = 2, 3, 4, \dots, n \\ 1 & \text{dla } i = 1 \end{cases}$$

$$S = \begin{cases} 1 & \text{dla } i = 1 \\ S + i & \text{dla } i = 2, 3, 4, \dots, n \end{cases}$$

4. Schemat blokowy



5. Sprawdzenie dla $n=3$

1. start

2. $n=3$

3. $S=0$ $i=1$

4. $i \leq n$ $1 \leq 3$ TAK

5. $S=0+1=1$
 $i=1+1=2$

4. $i \leq n$ $2 \leq 3$ TAK

5. $S=1+2=3$
 $i=2+1=3$

4. $3 \leq 3$ TAK

5. $S=3+3=6$
 $i=3+1=4$

4. $i \leq n$ $4 \leq 3$ NIE

6. $S(3)=6$

STOP

SUMA rekurencyjnie

1. dane użadone

$$\{a_1, a_2, a_3, a_4, \dots, i, \dots, (n-4)(n-3)(n-2)(n-1), n\} =$$

$$\{1, 2, 3, 4, \dots, i, \dots, (n-4)(n-3)(n-2)(n-1), n\}, \text{ gdzie } n \in \mathbb{N}$$

2. Określanie nymik

$$S(n) = \sum_{i=1}^n i$$

3. Analiza rekurencja:

$$S(n) = S(n-1) + n$$

$$S(n-1) = S(n-2) + n-1$$

$$S(n-2) = S(n-3) + n-2$$

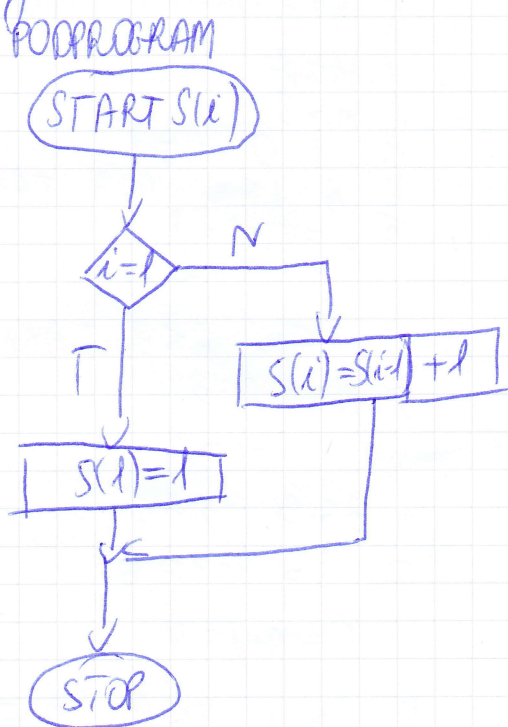
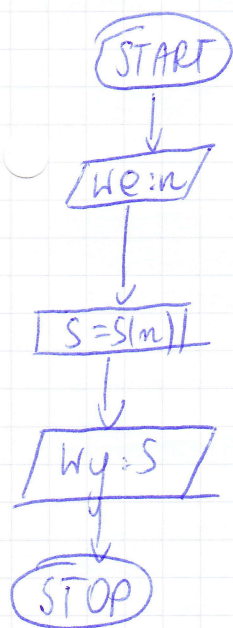
$$S(i) = S(i-1) + i$$

$$S(1) = a_1$$

$$S(i) = \begin{cases} 1 & \text{dla } i=1 \\ S(i-1) + i & \text{dla } i=n, n-1, n-2, n-3, \dots \end{cases}$$

$$S = \begin{cases} 1 & \text{dla } i=1 \\ S+i & \text{dla } i=n, n-1, n-2, n-3, \dots \end{cases}$$

4. Schemat blokowy



5. Sprawdzenie dla $n=3$

$$S(3) = S(2) + 3 = 3 + 3 = 6$$

$$S(2) = S(1) + 2 = 1 + 2 = 3$$

$$S(1) = 1$$

Algorytm
dłata
wprawmie

Suma dwóch parytych iteracyjnie

1. Dane uładowe

$$\{a_1, a_2, a_3, \dots, i, \dots, (n-2), (n-1), n\} = \{2, 4, 6, \dots, i, \dots, (n-2)(n-1)(n)\}$$

2. Oczekiwany wynik

$$S(n) = \sum_{i=1}^n 2i$$

3. Analiza zadania

$$S(1) = a_1$$

$$S(n) = S(n-1) + 2n$$

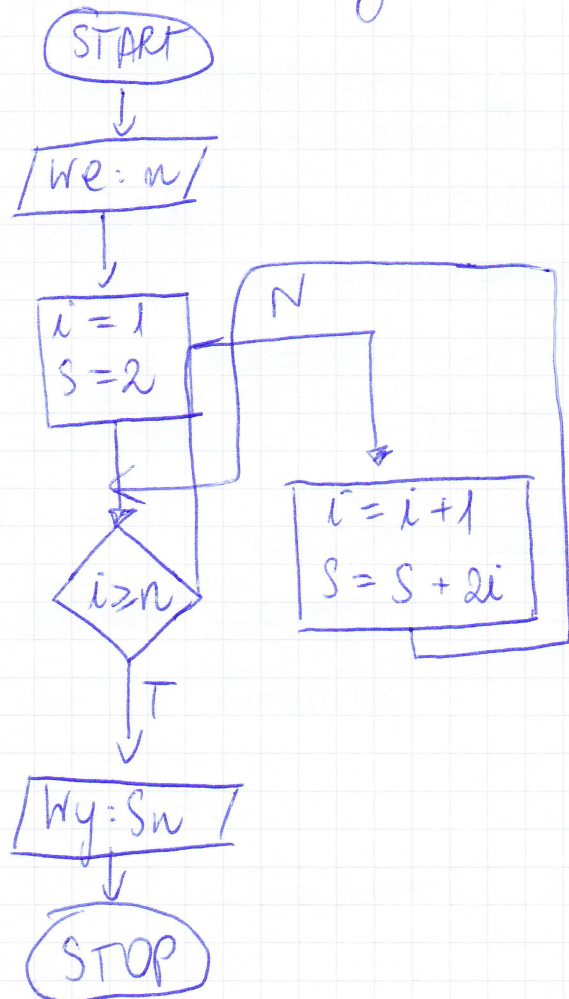
$$S(n-1) = S(n-2) + 2n$$

$$S(n-2) = S(n-3) + 2n$$

$$S(i) = S(i-1) + 2i$$

$$S(i) = \begin{cases} 2 & \text{dla } i=1 \\ S(i-1) + 2i & \text{dla } i=2, 3, 4, \dots, n \end{cases}$$

4. Schemat blokowy



5. sprawdzanie dla $n=3$

1. start

2. $n=3$

3. $i=1$ $S=2$

4. $i \geq n$ $1 \geq 3$ nie

5. $i = 1+1 = 2$

$$S = 2 + 4 = 6$$

4. $i \geq n$ $2 \geq 3$ NIE

5. $i = 2+1 = 3$

$$S = 6 + 6 = 12$$

4. $i \geq n$ $3 \geq 3$ TAK

$$S(3) = 12$$

✓ mi oszczędza algorytm danych poprawnie

Suma liczb parzystych rekurencyjnie

1. Dane wsadowe:

$$\{a_1, a_2, a_3, \dots, i, \dots, (n-2), (n-1), n\} = \{2, 4, 6, \dots, i, \dots, (n-2), (n-1), n\}$$

gdzie $n \in \mathbb{N}$

2. Owekowany mymik

$$S(n) = \sum_{i=1}^n (2i)$$

3. Analiza zadania

$$S(n) = S(n-1) + 2n$$

$$S(n-1) = S(n-2) + 2(n-1)$$

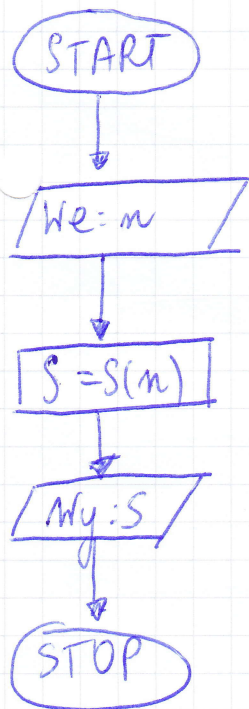
$$S(n-2) = S(n-3) + 2(n-2)$$

$$S(i) = S(i-1) + 2i$$

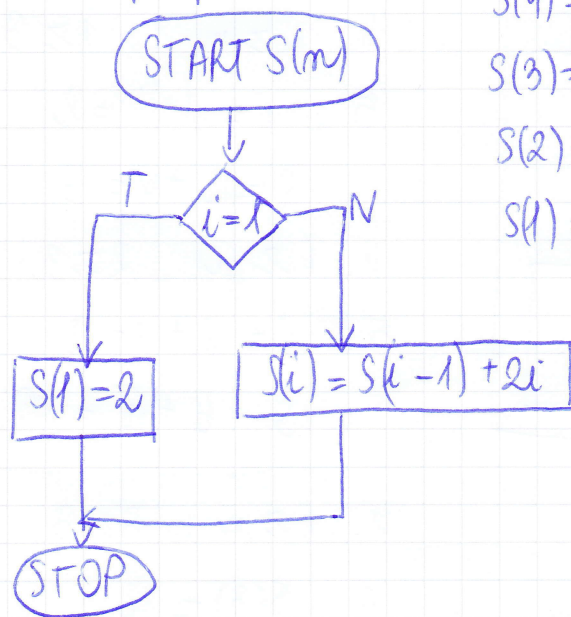
$$S(1) = a_1$$

$$S(i) = \begin{cases} S(i-1) + 2i - 1 & \text{dla } i = n, (n-1), (n-2), \dots, 3 \\ 2 & \text{dla } i = 2 \\ 1 & \text{dla } i = 1 \end{cases}$$

4. Schemat blokowy



Podprogram



5. Sprawdzenie dla $n=4$

$$S(4) = S(3) + 8 = 12 + 8 = 20$$

$$S(3) = S(2) + 6 = 6 + 6 = 12$$

$$S(2) = S(1) + 4 = 2 + 4 = 6$$

$$S(1) = 2$$

Wnioski:

Algorytm działa poprawnie

suma nieparzystych iteracyjnie

1. Dane n i a₁

$$\{a_1, a_2, a_3, \dots, i, \dots, n\} = \{1, 3, 5, \dots, i, \dots, n\}$$

2. Obliczamy wyniki

$$S(n) = \sum_{i=1}^n (2i-1)$$

3. Analiza rekurencji

$$S(1) = a_1$$

$$S(2) = a_1 + a_2$$

$$S(3) = S(2) + a_3$$

$$S(4) = S(3) + a_4$$

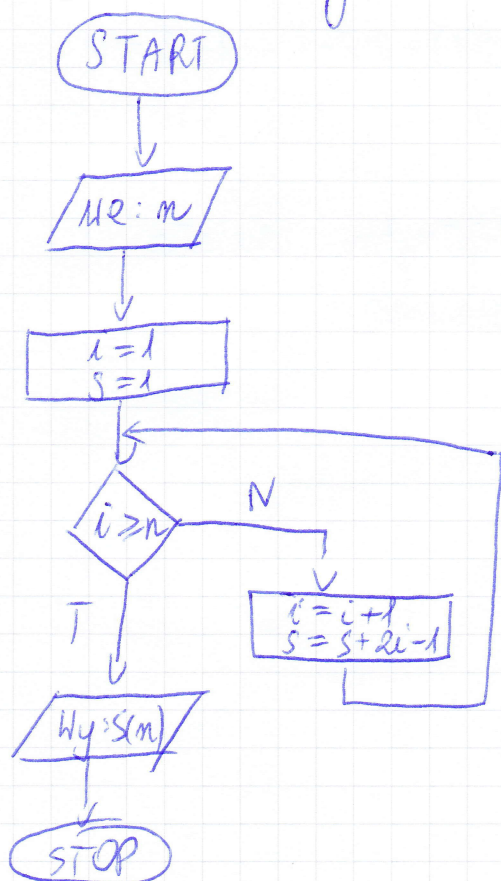
$$S(i) = S(i-1) + 2i-1$$

$$S(n) = S(n-1) + 2n-1$$

$$S(i) = \begin{cases} 1 & \text{dla } i=1 \\ S(i-1) + 2i-1 & \text{dla } i=2, 3, 4, \dots, n \end{cases}$$

$$S = \begin{cases} 1 & \text{dla } i=1 \\ S + 2i-1 & \text{dla } i=2, 3, 4, \dots \end{cases}$$

Schemat blokowy



Sprawdzenie dla $n=2$

1. start

2. $n=2$

3. $i=1$ $S=1$

4. $i \geq n$ $1 \geq 2$ nie

5. $i = 1+1 = 2$

$$S = 1 + 2 \cdot 2 - 1 = 4$$

4. $2 \geq 2$ TAK

6. $S(2) = 4$

Algorytm działa poprawnie

Algorytm nr. 1

suma liczb nieparzystych rekurencyjnie

1. Dane wejściowe:

$\{a_1, a_2, a_3, \dots, i, \dots, (n-2), (n-1), (n)\} = \{1, 3, 5, \dots, i, (n-2)(n-1)(n)\}$
gdzie $n \in \mathbb{N}$.

2. Określamy wyniki:

$$S(n) = \sum_{i=1}^n (2i-1)$$

3. Analiza rekurencja:

$$S(n) = S(n-1) + 2n-1$$

$$S(n-1) = S(n-2) + 2n-1$$

$$S(n-2) = S(n-1) + 2n-1$$

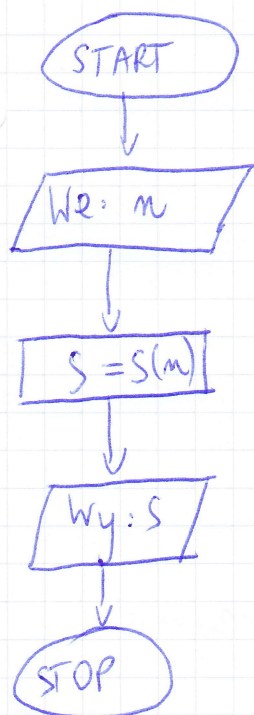
$$S(i) = S(i-1) + 2i-1$$

$$S(1) = a_1$$

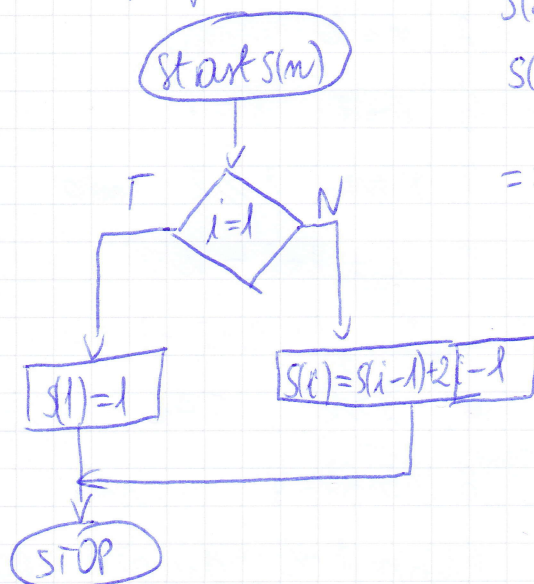
$$S(i) = \begin{cases} S(i-1) + 2i-1 & \text{dla } i = n, (n-1), n-2, \dots, 2 \\ 1 & \text{dla } i = 1 \end{cases}$$

$$S = \begin{cases} S + 2i-1 & \text{dla } i = n, n-1, n-2, \dots, 2 \\ 1 & \text{dla } i = 1 \end{cases}$$

4. Schemat blokowy



Podprogram



5. Sprawdzenie dla $n=3$

$$S(3) = S(2) + 2 \cdot 3 - 1 = 4 + 6 - 1 = 9$$

$$S(2) = S(1) + 2 \cdot 2 - 1 = 1 + 3 = 4$$

$$S(1) = 1$$

$\Rightarrow$ algorytm działa poprawnie

1. LÓŻYMY ITERACJIE

1. dane usadów

$$\{a_1, a_2, a_3, \dots, i, \dots, m\} = \{1, 2, 3, \dots, i, \dots, n\}, \text{ gdzie } m \in \mathbb{N}$$

2. Określamy myślik

$$I(i) = a_1 \cdot a_2 \cdot a_3 \cdot \dots \cdot i \cdot \dots \cdot m$$

3. Analiza zadania

$$I(1) = 1$$

$$I(2) = I(1) \cdot 2$$

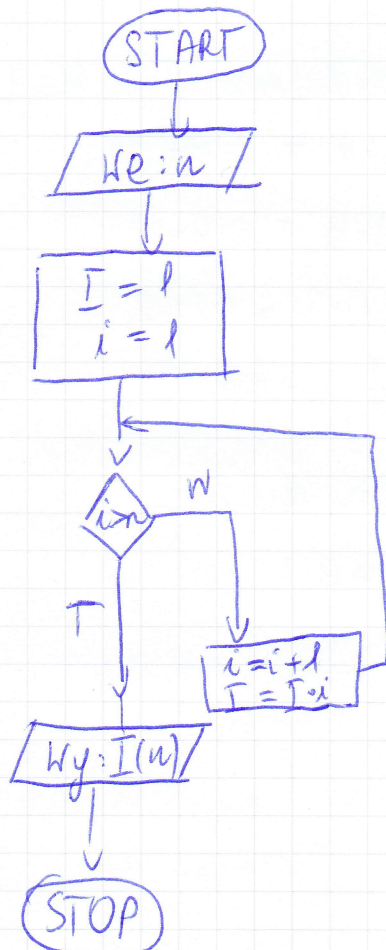
$$I(3) = I(2) \cdot 3$$

$$I(4) = I(3) \cdot 4$$

$$I(i) = \begin{cases} I(m-1) \cdot i & \text{dla } i = 2, 3, \dots, m \\ 1 & \text{dla } i = 1 \end{cases}$$

$$I = \begin{cases} I \cdot i & \text{dla } i = 2, 3, \dots, m \\ 1 & \text{dla } i = 1 \end{cases}$$

4. Schemat blokowy



5. Sprawdzenie dla $n=3$

1. START

2. $n = 3$

3. $I = 1$ $i = 1$

4. $i > n$ $1 > 3$ NIE

5. $i = 1 + 1 = 2$

$$I = 1 \cdot 1 = 1$$

4. $i > n$ $3 > 3$ NIE

5. $i = 3 + 1 = 4$

$$I = 2 \cdot 3 = 6$$

4. $i > n$ $4 > 3$ TAK

6. $I(3) = 6$

7. Stop

WNIOSK: Algorytm działa poprawnie

1. LOCZYN REKURENCYJNIE

1. Dane wiadome:

$$\{a_1, a_2, a_3, \dots, i, \dots, (n-2)(n-1), n\} = \{1, 2, 3, \dots, i, \dots, (n-2)(n-1), n\}$$

gdzie $n \in \mathbb{N}$

2. Oczekiwany wynik

$$I(n) = a_1 \cdot a_2 \cdot a_3 \cdot \dots \cdot i \cdot \dots \cdot (n-2)(n-1) \cdot n$$

3. Analiza rekurencja

$$I(n) = I(n-1) \cdot n$$

$$I(n-1) = I(n-2) \cdot (n-1)$$

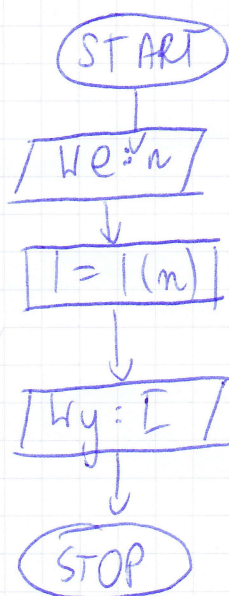
$$I(n-2) = I(n-3) \cdot (n-2)$$

$$I(i) = I(i-1) \cdot i$$

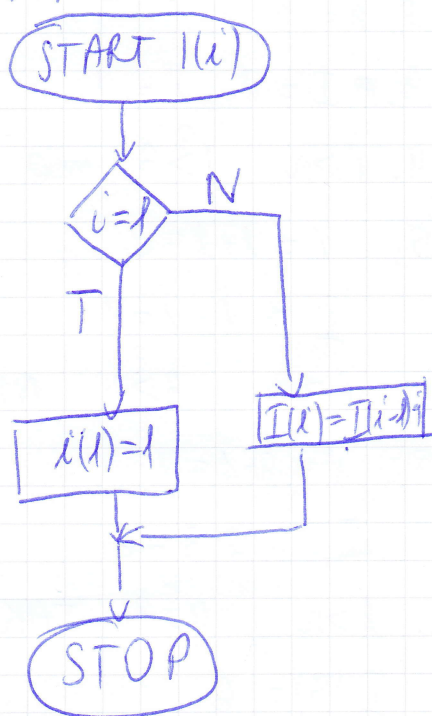
$$I(1) = 1$$

$$I(i) = \begin{cases} I(i-1) \cdot i & \text{dla } i = n, n-1, n-2, \dots, 2 \\ 1 & \text{dla } i = 1 \end{cases} \quad I = \begin{cases} i & \text{dla } i = 2, 3, \dots, n \\ 1 & \text{dla } i = 1 \end{cases}$$

4. Schemat blokowy



PODPROGRAM



5. Sprawdzenie dla $n=4$

$$I(4) = I(3) \cdot 4 = 6 \cdot 4 = 24$$

$$I(3) = I(2) \cdot 3 = 2 \cdot 3 = 6$$

$$I(2) = I(1) \cdot 2 = 1 \cdot 2 = 2$$

$$I(1) = 1$$

Algorytm działa poprawnie

Wzrost i obrotowy iteracyjny

1. Dane wejściowe

$a_1, a_2, a_3, \dots, i, \dots, (n-2)(n-1)(n) = \{2, 4, 6, \dots, i, \dots, (n-2)(n-1)(n)\}$

2. Obliczenia

$$I(n) = a_1 \cdot a_2 \cdot a_3 \cdot \dots \cdot i \cdot \dots \cdot (n-2)(n-1) \cdot n$$

3. Analiza rekurencyjna

$$I(4) = 2 \cdot 4 \cdot 6 \cdot 8$$

$$I(4) = I(3) \cdot 8$$

$$I(n) = I(n-1) \cdot 2n$$

$$I(3) = 2 \cdot 4 \cdot 6$$

$$I(3) = I(2) \cdot 6$$

$$I(n-1) = I(n-2) \cdot 2(n-1)$$

$$I(2) = 2 \cdot 4$$

$$I(2) = I(1) \cdot 4$$

$$I(n-2) = I(n-3) \cdot 2(n-2)$$

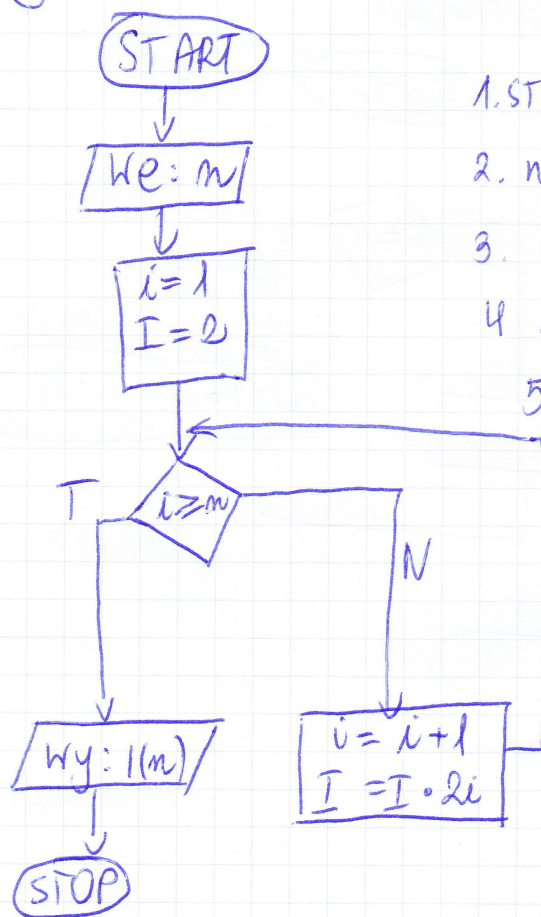
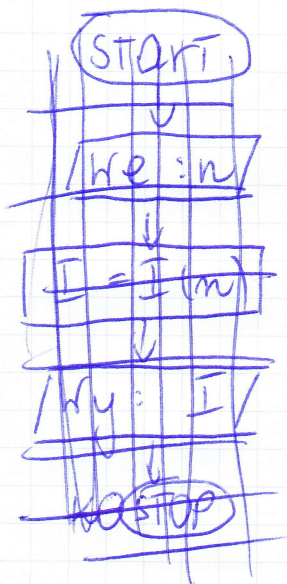
$$I(1) = 2$$

$$I(1) = 2$$

$$I(1) = 2$$

$$I(i) = \begin{cases} I(i-1) \cdot 2i & \text{dla } i = 2, 3, 4, \dots, n \\ 2 & \text{dla } i = 1 \end{cases}$$

4. Schemat blokowy



5. Sprawdzenie dla n=3

1. START

2. n=3

3. i=1 I=2

4. i ≥ n 1 ≥ 3 NIE

5. i = 1 + 1 = 2

I = 2 · 4 = 8

4. 2 ≥ 3 NIE

5. i = 2 + 1 = 3

I = 8 · 6 = 48

4. 3 ≥ 3 TAK

6. I(3) = 48

7. STOP

Wniosek: algorytm działa poprawnie

1. Wypisać parytyczny rekurencyjnie

1. Dane masowe

$$\{a_1, a_2, a_3, \dots, i, \dots, (n-2)(n-1)n\} = \{2, 4, 6, \dots, i, \dots, (n-2)(n-1)n\}$$

2. Określony wynik

$$I(n) = a_1 \cdot a_2 \cdot a_3 \cdot \dots \cdot i \cdot \dots \cdot (n-2)(n-1)(n)$$

3. Analiza rekurencji

$$I(3) = 2 \cdot 4 \cdot 6$$

$$I(n) = I(n-1) \cdot 2n$$

$$I(2) = 2 \cdot 4$$

$$I(n-1) = I(n-2) \cdot 2(n-1)$$

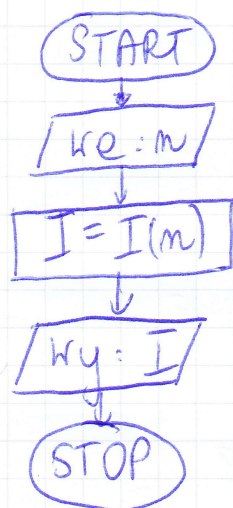
$$I(1) = 2$$

$$I(1) = 2$$

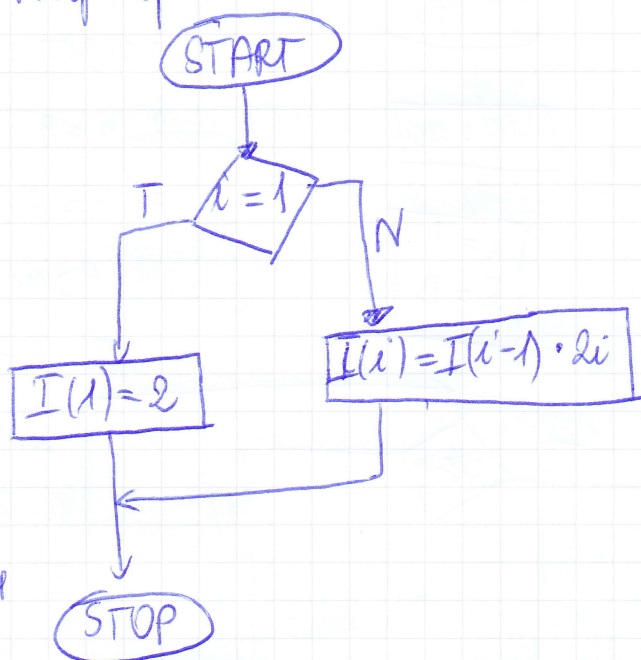
$$I(i) = I(i-1) \cdot 2i$$

$$I(i) = \begin{cases} I(i-1) \cdot 2i & \text{dla } i = 2, 3, 4, \dots, (n-2)(n-1)(n) \\ 2 & \text{dla } i = 1 \end{cases}$$

4. Schemat blokowy



Podprogram



5. Sprawdzenie dla $n=4$

$$I(4) = I(3) \cdot 8 = 48 \cdot 8 = 384$$

$$I(3) = I(2) \cdot 6 = 8 \cdot 6 = 48$$

$$I(2) = I(1) \cdot 4 = 8$$

$$I(1) = 2$$

Wniosek: algorytm działa poprawnie

Wzrost liczby parzystych iteracji

1. dane i sadowne

$\{a_1, a_2, a_3, \dots, i, \dots, (n-2)(n-1)(n)\} = \{1, 3, 5, \dots, (n-2)(n-1)(n)\}$
gdzie $n \in \mathbb{N}$

2. One liwary by mih

$$I(n) = a_1 \cdot a_2 \cdot a_3 \cdot \dots \cdot i \cdot \dots \cdot (n-2)(n-1) \cdot (n)$$

3. Analize sadowne

$$I(n) = I(n-1) \cdot (2n-1)$$

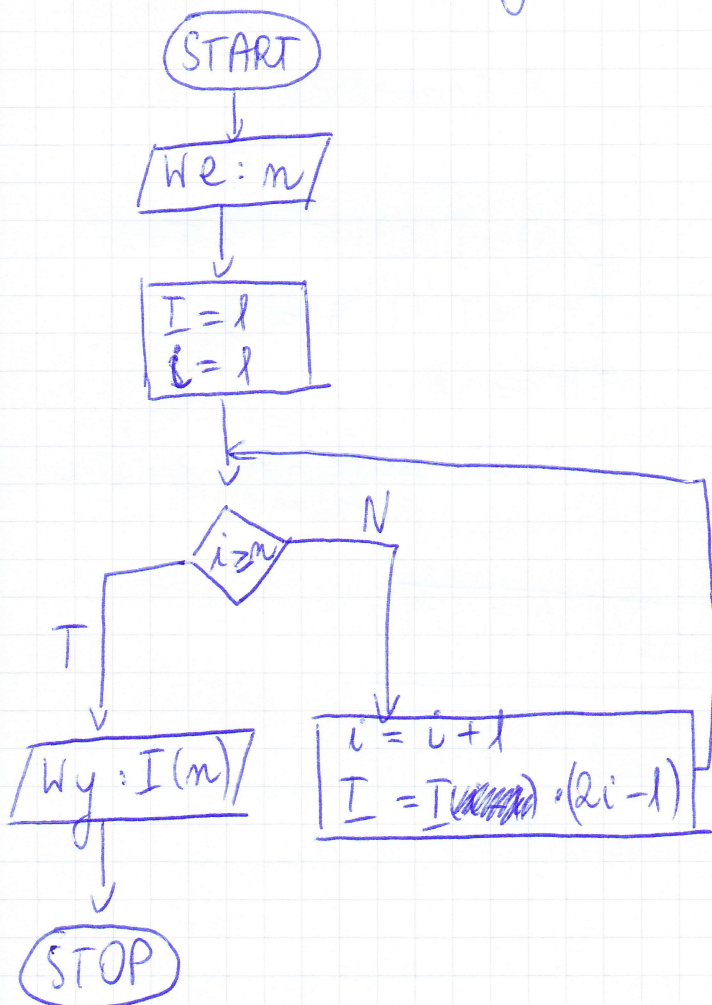
$$I(n-1) = I(n-2) \cdot (2n-1)$$

$$I(i) = I(i-1)(2i-1)$$

$$I = \prod_{i=1}^n I(i-1) \cdot (2i-1) \text{ dla } i = 2, 3, \dots, n$$

4. schemat blokowy:

5. sprawdzanie dla $n=3$



1. START

2. $n=3$

3. $I=1$

$i=1$

4. $1 \geq 3$ NIE

5. $i = 1 + 1 = 2$

$I = 1 \cdot 3 = 3$

4. $2 \geq 3$ NIE

5. $i = 2 + 1 = 3$

$I = 3 \cdot 5 = 15$

5. 4. $3 \geq 3$ TAK

~~$i = 3 + 1 = 4$~~

6. $I(3) = 15$

~~$I = 15 \cdot 7 = 105$~~

7. STOP

~~$4 \cdot 4 \geq 3$~~

~~TAK~~

~~6. $I(3) = 105$~~

Wniosek:
algorytm działa poprawnie

1. Wzrost liczby nieparzystych rekurencyjnie

1. Podane wsadzone

$$\{a_1, a_2, a_3, \dots, i, \dots, (n-2), (n-1), (n)\} = \{1, 3, 5, \dots, i, \dots, (n-2), (n-1), (n)\}$$

gdzie $n \in \mathbb{N}$

2. Oczekiwany wynik

$$I(n) = a_1 \cdot a_2 \cdot a_3 \cdot \dots \cdot i \cdot \dots \cdot (n-2) \cdot (n-1) \cdot (n)$$

3. Analiza rekurencja

$$I(4) = 1 \cdot 3 \cdot 5 \cdot 7$$

$$I(3) = 1 \cdot 3 \cdot 5$$

$$I(2) = 1 \cdot 3$$

$$I(1) = 1$$

$$I(n) = I(n-1) \cdot (2n-1)$$

$$I(n-1) = I(n-2) \cdot (2n-1)$$

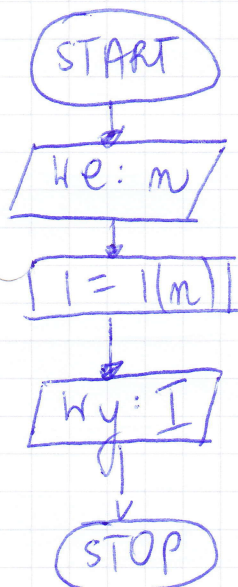
$$I(n-2) = I(n-3) \cdot (2n-1)$$

$$I(1) = 1$$

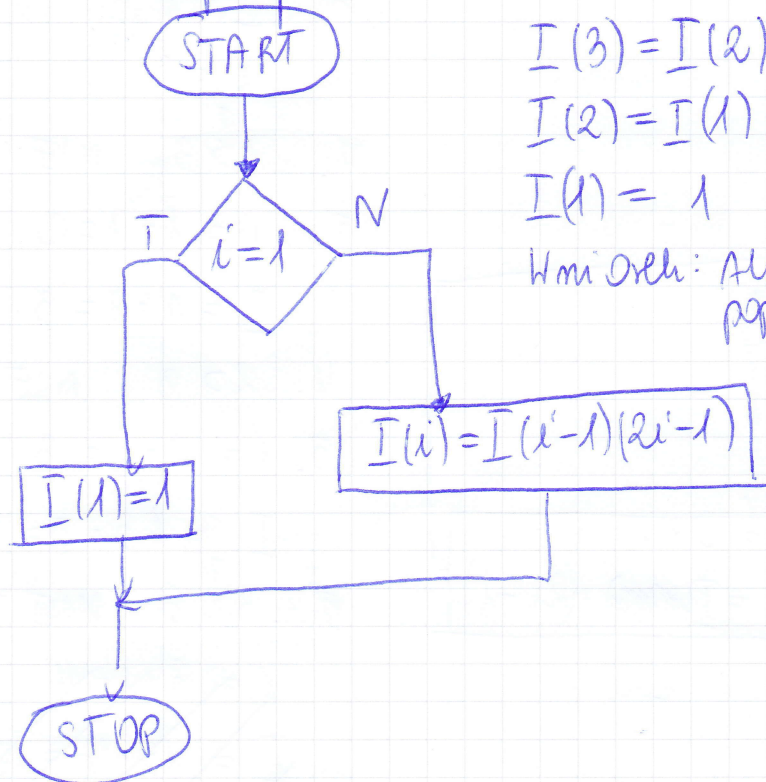
$$I(i) = I(i-1) \cdot (2i-1)$$

$$I(i) = \begin{cases} I(i-1) \cdot (2i-1) & \text{dla } i = 2, \dots, (n-2), (n-1), (n) \\ 1 & \text{dla } i = 1 \end{cases}$$

4. Schemat blokowy



Podprogram



5. Sprawdzenie dla $n=3$

$$I(3) = I(2) \cdot 5 = 3 \cdot 5 = 15$$

$$I(2) = I(1) \cdot 3 = 3$$

$$I(1) = 1$$

Wniosek: Algorytm działa poprawnie

POTĘGA ITERACYJNIE

1. Dane miodowe

$\{a_1, a_2, a_3, \dots, i, \dots, n\} = \{0, 1, 2, \dots, i, \dots, n\}$ gdzie $n \in \mathbb{N} \cup \{0\}$

2. Określamy symbol

$$W(n) = a^n$$

3. Analiza rekurencji

$$W(0) = 1$$

$$W(1) = W(0) \cdot a = a$$

$$W(2) = W(1) \cdot a = a^2$$

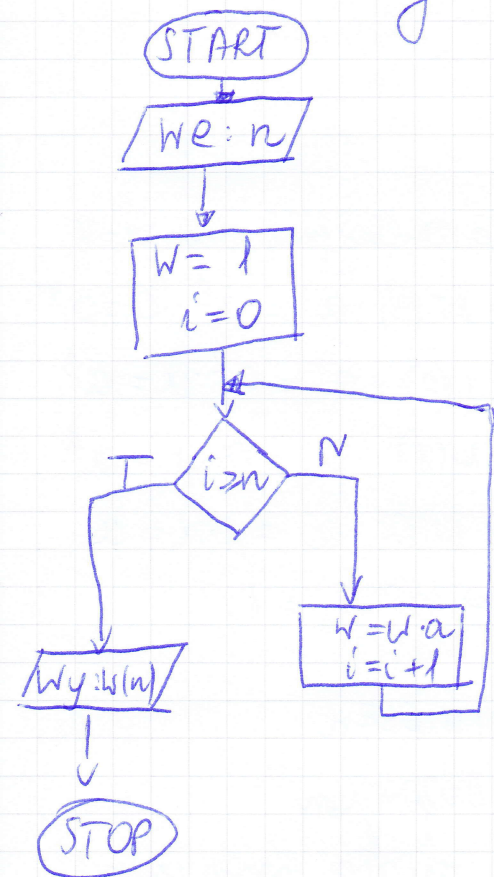
$$W(i) = W(i-1) \cdot a = a^i$$

$$W(n) = W(n-1) \cdot a = a^n$$

$$W(n) = \begin{cases} 1 & \text{dla } i=0 \\ W(i-1) \cdot a & \text{dla } i=1, 2, 3, \dots, n \end{cases}$$

$$W = \begin{cases} 1 & \text{dla } i=0 \\ a^i & \text{dla } i=1, 2, 3, \dots, n \end{cases}$$

4. Schemat blokowy



5. Sprawdzenie dla $n=1$

1. START

2. $n=1$

3. $W=1$ $i=1$

4. $0 \geq 1$ NIE

5. $W = W \cdot a = a$

$i = 0 + 1 = 1$

6. $i \geq n$

$1 \geq 1$ TAK

7. $W = a$

8. STOP

Wniosek: Algorytm DZIAŁA
POPRAWNIE

POTĘGA REKURENCYJNIE

1. Dane n i a dane

$$\{a_1, a_2, a_3, \dots, i, \dots, (n-2), (n-1), n\} = \{0, 1, 2, 3, \dots, i, \dots, (n-2), (n-1), n\}$$

gdzie $n \in \mathbb{N} \cup \{0\}$

2. Oczekiwany wynik

$$w(n) = a^n$$

3. Analiza rekurencji

$$w(n) = w(n-1) \cdot a$$

$$w(n-1) = w(n-2) \cdot a$$

$$w(n-2) = w(n-3) \cdot a$$

$$w(i) = w(i-1) \cdot a$$

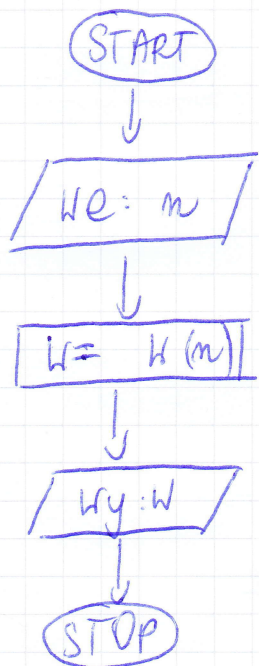
$$w(1) = w(0) \cdot a$$

$$w(0) = a^0 \cdot 1$$

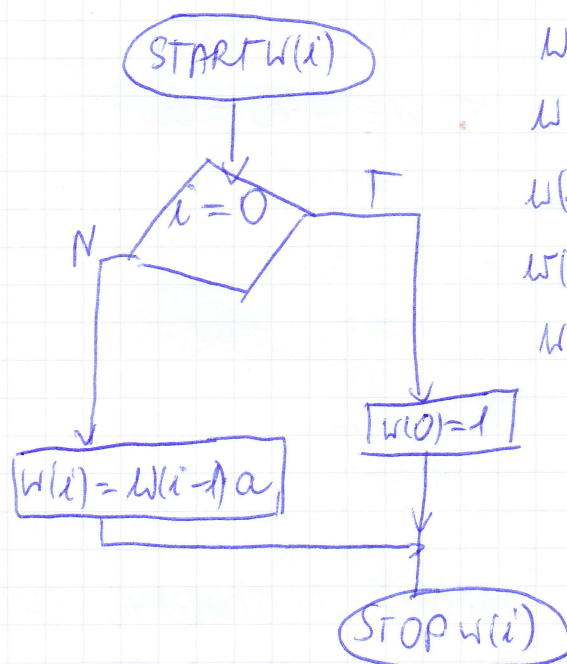
$$w(i) = \begin{cases} w(i-1) \cdot a & \text{dla } i = n, n-1, n-2, \dots, 1 \\ 1 & \text{dla } i = 0 \end{cases}$$

$$w = \begin{cases} a^i & \text{dla } i = n, n-1, n-2, \dots, 1 \\ 1 & \text{dla } i = 0 \end{cases}$$

4. Schemat blokowy



POPPROGRAM



5. Sprawdzanie dla $n=4$

$$w(4) = w(3) \cdot a = a^3 \cdot a = a^4$$

$$w(3) = w(2) \cdot a = a^2 \cdot a = a^3$$

$$w(2) = w(1) \cdot a = a \cdot a = a^2$$

$$w(1) = w(0) \cdot a = 1 \cdot a = a$$

$$w(0) = a^0 \cdot 1 = 1$$

WNIOSEK

Algorytm

działa poprawnie