zadanie 2 AiP konserwatoria

spis treści

zadanie 1

1a

złożoność:

> n operacji == Θ(n)

dane:

> a - lista liczb
>
> n - dlugosc listy a

pseudokod:

```
suma <- 0
dla i <- 1 do n wykonaj
    suma <- suma+a[i]
```

python:

```
suma=0
for i in range(n): #w przypadku pythona trzeba pamiętać że listę iterujemy od zera
    suma+=a[i]
```

1b

złożoność:

> n operacji == Θ(n)

dane:

> a - lista liczb
>
> n - dlugosc listy a

pseudokod:

```
iloczyn <- 1
dla i <- 1 do n wykonaj
    iloczyn <- iloczyn*a[i]
```

python:

```
iloczyn = 1
for i in range(n): #iterujemy od zera
    iloczyn*=a[i]
```

1c

złożoność:

> n+1 operacji == Θ(n)

dane:

> a - lista liczb
>
> n - dlugosc listy a

pseudokod:

```
suma <- 0
dla i <- 1 do n wykonaj
    suma <- suma+a[i]
srednia = suma / n
```

python:

```
suma=0
for i in range(n): #iterujemy od zera
    suma+=a[i]
srednia = suma / n
```

1d

złożoność:

> n+1 operacji == Θ(n)

dane:

> a - lista liczb
>
> n - dlugosc listy a

pseudokod:

```
suma <- 0
ilosc <- 0
dla i <- 1 do n wykonaj
jesli a[i] większa od 0 to
    suma <- suma+a[i]
    ilosc <- ilosc + 1
srednia <- suma/ilosc
```

python:

```
suma=0
ilosc=0
for i in range(n): #iterujemy od zera
    if a[i]>0:
        suma+=a[i]
        ilosc+=1
srednia = suma/ilosc
```

1e

złożoność:

> (n^2+n)/2 operacji == Θ(n^2)

dane:

> a - lista liczb
>
> n - dlugosc listy a

pseudokod:

```
suma <- 0
dla k <- 1 do n wykonaj
    iloczyn <- 1
    dla i <- 1 do k wykonaj
        iloczyn <- iloczyn*a[i]
    suma <- suma+iloczyn
```

python:

```
suma=0
for k in range(n): #iterujemy od zera
    iloczyn = 1
    for i in range(k):
        iloczyn*=a[i]
    suma+=iloczyn
```

1f

złożoność:

> (1n/2)*n operacji == Θ(n^2)

dane:

> a - lista liczb
>
> n - dlugosc listy a

pseudokod:

```
suma <- 0
dla k <- 1 do n wykonaj
    iloczyn <- 1
    dla i <- k do n wykonaj
        iloczyn <- iloczyn*a[i]
    suma <- suma+iloczyn
```

python:

```
suma=0
for k in range(n): #iterujemy od zera
    iloczyn = 1
    for i in range(k, n):
        iloczyn*=a[i]
    suma+=iloczyn
```

zadanie 2

2a

analiza: 12, 6, 3, 10, 5, 16, 8, 4, 2, 1

wynik: 10x liczba musi przejsc przez szarą ramke

2b

dane:

> n - liczba całkowita

pseudokod:

```
ZAGATKA(n):
    ilosc <- 0
    dopóki Prawda
        ilosc <- ilosc+1
        jeżeli n modulo 2 porównywalne z 0
            n <- n/2
        w innym wypadku:
            jeżeli n porównywalne z 1:
                zakończ pętlę
            w innym wypadku:
                n <- n*3+1
    zwróć ilosc
```

python:

```
def ZAGATKA(n):
    ilosc=0
    while True:
        ilosc+=1
        if n%2 == 0:
            n//=2
        else:
            if n == 1:
                break
            else:
                n=n*3+1
    return ilosc
```

zadanie 3

3 dodawanie macierzy

złożoność:

> n2 operacji == Θ(n2)

dane:

> m - szerokość obydwu macierzy
>
> n - wysokość obydwu macierzy
>
> A - macierz 1
>
> B - macierz 2
>
> c - macierz wynikowa (wynik działania)

pseudokod:

```
dla i <- 1 do n wykonaj
    dla j <- 1 do m wykonaj
        C[i][j] <- A[i][j] + B[i][j]
```

3 odejmowanie macierzy

złożoność:

> n2 operacji == Θ(n2)

dane:

> m - szerokość obydwu macierzy
>
> n - wysokość obydwu macierzy
>
> A - macierz 1
>
> B - macierz 2
>
> c - macierz wynikowa (wynik działania)

pseudokod:

```
dla i <- 1 do n wykonaj
    dla j <- 1 do m wykonaj
        C[i][j] <- A[i][j] - B[i][j]
```

3 mnożenie macierzy

złożoność:

> n2 operacji == Θ(n2)

dane:

> m - szerokość macierzy 1
>
> n - wysokość macierzy 1, szerokosc macierzy 2
>
> o - wysokość macierzy 2
>
> A - macierz 1
>
> B - macierz 2
>
> c - macierz wynikowa (wynik działania)

pseudokod:

```
dla i <- 1 do m wykonaj
    dla j <- 1 do o wykonaj
        suma <- 0
        dla k <- 1 do n wykonaj
            suma <- suma + A[i][k]*B[k][j]
        C[i][j] <- suma
```