

Opis funkcji:

Dana jest funkcja Pow¹:

```

1 def Pow(a, k):
2     z = a
3     y = 1
4     m = k
5     while m != 0:
6         if m % 2 == 1:
7             y = y*z
8
9         m = int(m/2)
10        z = z*z
11
12    return y

```

Gdzie k jest wartością całkowitą nieujemną zapisaną na n bitach, zatem:

$$k \in \langle 0; 2^n - 1 \rangle \wedge k \in \mathbb{Z}$$

1. Obliczyć pesymistyczną złożoność czasową przyjmując operację porównania w wierszu 6 funkcji jako operację dominującą.

Operacja porównania w wierszu 6 jest wywoływana podczas każdej iteracji pętli w wierszu 5. Podczas każdej iteracji tej pętli wartość m jest dzielona przez 2 bez reszty w wierszu 9. Gdy wartość m staje się równa 0 pętla przestaje się wykonywać. Na przykład:

15 → 7	9 → 4	17 → 8
7 → 3	4 → 2	8 → 4
3 → 1	2 → 1	4 → 2
1 → 0	1 → 0	2 → 1
		1 → 0

Liczby 9 i 15 da się zapisać na 4 bitach, ale liczbę 17 da się zapisać na minimalnie 5 bitach. Zatem pętla wykonuje się dla danego k n_m razy, gdzie n_m to minimalna liczba bitów potrzebna do zapisania liczby k .

Zatem pesymistyczna złożoność dla $n \in \mathbb{N}^+$ to:

$$T_{\text{pes}}(n) = n$$

2. Obliczyć średnią złożoność czasową przyjmując operację porównania w wierszu 6 funkcji jako operację dominującą.

Z poprzedniego zadania wiemy że złożoność dla liczby zapisanej na minimalnej liczbie bitów jest równa ilości jej bitów. Zatem złożoność średnia jest równa:

¹tu przepisana na język *Python* z zachowaniem numeracji wierszy z danego zapisu w pseudokodzie

$$\begin{aligned}
T_{\text{sr}}(n) &= \frac{1 \cdot 1 + 2 \cdot 2 + 3 \cdot 4 + 4 \cdot 8 + \dots + n \cdot 2^{n-1}}{2 + 2 + 4 + 8 + \dots + 2^{n-1}} = \\
&= \frac{1 \cdot 1 + 2 \cdot 2 + 3 \cdot 4 + 4 \cdot 8 + \dots + n \cdot 2^{n-1}}{2^n} = \\
&= \frac{2^n - 1 + 2(2^n - 1) + 4(2^n - 1) + \dots + 2^{n-1} \cdot 1}{2^n} = \\
&= \frac{2^n - 1 + 2^n - 2 + 2^n - 4 + \dots + 2^n - 2^{n-1}}{2^n} = \\
&= \frac{n \cdot 2^n - (2^n - 1)}{2^n} = \frac{(n - 1) \cdot 2^n + 1}{2^n} = \\
&= n - 1 + \frac{1}{2^n}
\end{aligned}$$

3. Obliczyć średnią liczbę wykonanych operacji mnożenia w wierszu 7 funkcji

Przyjrzyjmy się warunkowi w wierszu 6 funkcji, warunek $m \% 2 == 1$ musi zostać spełniony by mnożenie w wierszu 7 zostało wykonane. Warunek ten jest spełniony tylko i wyłącznie wtedy gdy liczba jest nieparzysta, czyli ostatni jej bit jest równy 1.

Zmienna m jest dzielona przez 2 bez reszty podczas każdej iteracji pętli w wierszu 5. Zobaczmy co się dzieje podczas tej operacji z liczbą m :

16 : 10000	22 : 10110
8 : 1000	11 : 1011
4 : 100	5 : 101
2 : 10	2 : 10
1 : 1	1 : 1

Podczas dzielenia przez 2 bez reszty bity są przesuwane w prawo, dlatego każda cyfra w zapisie binarnym liczby będzie tą *ostatnią* która zostanie wzięta pod uwagę w operacji porównywania w wierszu 6. Zatem mnożenie w wierszu 7 zostanie wykonane tyle razy, ile jedynek posiada liczba w zapisie binarnym.

Zatem należy obliczyć średnią liczbę jedynek w n bitowej liczbie całkowitej nieujemnej zapisanej w kodzie binarnym.

Niech a_n to ciąg łącznej ilości jedynek w liczbach n bitowych. Zaczniemy wypisywać po kolei kolejne liczby binarne począwszy od 0.

0, 1, 10, 11, ...

Możemy zauważyć iż liczba trzecia i liczba czwarta to odpowiednio liczba pierwsza i liczba druga, z tą różnicą że posiadają dodatkową jedynkę z przodu. Zatem ilość jedynek w tych czterech liczbach to dwukrotność ilości jedynek w dwóch pierwszych liczbach plus 2. Zatem możemy określić ciąg a_n rekurencyjnie:

$$\begin{cases} a_1 = 1 \\ a_n = 2 \cdot a_{n-1} + 2^n \end{cases}$$

Średnia ilość jedynek c_n w liczbach n bitowych jest zatem równa:

$$\begin{aligned}c_n &= \frac{2 \cdot a_{n-1} + a^{n-1}}{2^n} = \\&= \underbrace{\frac{a_{n-1}}{2^{n-1}}}_{c_{n-1}} + \frac{1}{2} = c_{n-1} + \frac{1}{2}\end{aligned}$$

Oraz:

$$c_1 = \frac{a_1}{2^1} = \frac{1}{2}$$

Ciąg c_n jest arytmetyczny ponieważ różnica kolejnych wyrazów jest stała, zatem:

$$c_n = \frac{1}{2} + (n-1) \cdot \frac{1}{2} = \frac{n}{2}$$

Więc instrukcja mnożenia w wierszu 7 jest wykonywana średnio $\frac{n}{2}$ razy, dla liczb całkowitych nieujemnych zapisanych na n bitach.

4. Opisać krótko, jaki warunek musi spełnić wykładnik k aby zachodził przypadek pesymistyczny

Z obliczeń z zadania pierwszego wiemy że złożoność dla liczby k_4 zapisanej na minimalnej liczbie bitów jest równa ilości jej bitów. Zatem liczba bitów k_4 jest równa n .

Więc k_4 musi spełniać warunek:

$$k_4 \in \langle 2^{n-1} - 1; 2^n - 1 \rangle \wedge k_4 \in \mathbb{Z}$$

aby wykładnik być przypadkiem pesymistycznym funkcji Pow.

5. Obliczyć liczbę przypadków, dla których zachodzi przypadek pesymistyczny

Z poprzedniego zadania wiemy, że przypadek pesymistyczny zachodzi dla liczby przypadków k_5 , która jest zależna od n

$$\begin{aligned}k_5 &= 2^n - 1 - (2^{n-1} - 1) = 2^n - 2^{n-1} = \\&= 2^{n-1}\end{aligned}$$