

Imię, nazwisko, indeks (można pominąć):

Zadanie 1. Poniżej podane zostały dwa algorytmy:**Algorytm 1**

1. $s \leftarrow 0$
2. dla $i = 1, 2, \dots, 50$ wykonaj:
 - a. $s \leftarrow s + a_i$
3. wypisz s

Algorytm 2

1. $s \leftarrow a_1$
2. dla $i = 2, 3, \dots, 50$ wykonaj:
 - a. Jeżeli $s < a_i$ to wykonaj:
 $s \leftarrow a_i$
3. wypisz s

Uwaga: zapis „ $x \leftarrow \text{wyrażenie}$ ” nadaje zmiennej x wartość równą podanemu wyrażeniu.**Odpowiedz na pytania:**

- Jaka wartość zostanie wypisana w punkcie 3. Algorytmu 1 i punkcie 3 Algorytmu 2 jeśli $a_i = 1$ dla wszystkich i nieparzystych i $a_i = 2$ dla wszystkich i parzystych?

Odp.:

Algorytm 1:.....

Algorytm 2:.....

- Jaka jest zależność między wartością s wypisywaną w wierszu 3. a liczbami $a_1, \dots, a_{50}$? Zależności możesz opisać przy pomocy wyrażeń matematycznych lub słownie.

Odp.:

Algorytm 1:

.....

Algorytm 2:.....

.....

Zadanie 2. Podaj algorytm, który wypisuje iloczyn największej i najmniejszej liczby w ciągu $a_1, \dots, a_{100}$. Algorytm możesz zapisać jako schemat blokowy, za pomocą wybranego języka programowania lub stosując notacje używane w algorytmach z Zadania 1.

Zadanie 3. Poniżej zapisana jest funkcja rekurencyjna Exp z parametrem naturalnym i :

Funkcja $\text{Exp}(i)$

1. Jeśli $i = 0$: zwróć 1 i zakończ działanie.
2. Jeśli (i nieparzyste) : zwróć $\text{Exp}(i - 1)$ i zakończ działanie.
3. Jeśli (i parzyste) : zwróć $\text{Exp}(i - 1) + \text{Exp}(i - 2)$ i zakończ działanie.

Twoje zadanie jest następujące:

- Uzupełnij poniższą tabelę:

i	0	1	2	3	4	5	6	7
$\text{Exp}(i)$	1	1						

- Podaj ogólną formułę określającą wartość zwracaną przez $\text{Exp}(i)$

Odp.:

.....

- Podaj ogólną formułę określającą liczbę wywołań („uruchomień”) $\text{Exp}(1)$ spowodowanych wywołaniem $\text{Exp}(i)$ dla dowolnej liczby naturalnej i .

Odp.:

.....

Zadanie 4. Złożoność czasowa algorytmów A_1 i A_2 rozwiązujących pewien problem P opisana jest odpowiednio zależnościami rekurencyjnymi T_1 i T_2 :

$$T_1(1) = 1$$

$$T_2(1) = 0$$

$$T_1(n) = n + T_1(n - 1) \text{ dla } n > 1$$

$$T_2(n) = 2 T_2(n / 2) + n \text{ dla parzystych } n > 1$$

$$T_2(n) = 2 T_2((n - 1) / 2) + n \text{ dla nieparzystych } n > 1$$

Który z algorytmów A_1 , A_2 szybciej rozwiązuje problem P dla dużych wartości n . Jeśli potrafisz, odpowiedź uzasadnij podając nierekurencyjną postać funkcji T_1 i T_2 dla argumentów n , które są naturalnymi potęgami dwójki.

Odp.:

Zadanie 5. Uzupełnij brakujące pola poniższej tabeli. Dla reprezentacji zmiennopozycyjnej przyjmij, że reprezentowana liczba jest równa $m \cdot 2^c$, mantysa (m) zajmuje 8 cyfr, cecha (c) zajmuje 4 cyfry, mantysa to liczba z przedziału $\langle 1,2 \rangle$ a cecha jest zapisana w kodzie uzupełnień do 2 (U2). Ponadto, $1 \leq m < 2$ oraz zapisujemy tylko cyfry „po przecinku” mantysy m .

Reprezentacja dziesiętna	Reprezentacja binarna stałopozycyjna	Reprezentacja binarna zmiennopozycyjna (cecha	
		mantysa	cecha
123			
	1110.11		
18.5	10010.1	00101000	0100
7.75			

Zadanie 6. Rozważmy zadanie polegające na sprawdzeniu, czy w uporządkowanym ciągu liczb $a_1 \leq a_2 \leq \dots \leq a_n$ znajduje się liczba x . Jeśli znasz ideę wyszukiwania binarnego,

- Opisz zastosowanie wyszukiwania binarnego dla powyższego; opis podaj jako schemat blokowy, w notacji z zadania 1, w wybranym języku programowania lub słownie.

Odp.:

- Podaj liczbę porównań, jakie opisany przez Ciebie algorytm wykona w najgorszym przypadku gdy $n = 10$, $n=1000$ i $n=10\,000$. Możesz przyjąć, że liczby $a_1, \dots, a_n$ są duuuużo większe od n , więc liczymy tylko porównania, w których biorą udział liczby z ciągu $a_1, \dots, a_n$ bądź ich „kopie”.

n	Liczba porównań
10	
1000	
10000	

Zadanie 7. Ustaw funkcje $f_1(n) = n^2$, $f_2(n) = n^{3/2}$, $f_3(n) = n \log n$, $f_4(n) = 2n$ w takiej kolejności, że jeśli $f_i(n)$ jest przed $f_j(n)$, to $f_i(n) = O(f_j(n))$. Jeśli nie rozumiesz polecenia, po prostu zostaw puste pole na odpowiedź.

Odp.:

Zadanie 8. Jeśli znasz algorytmy przeszukiwania w głąb i przeszukiwania wszerz, podaj przykładową kolejność, w jakiej *mogą być* odwiedzane wierzchołki poniższego grafu, jeśli startujemy z wierzchołka 1, a potem odwiedzamy wierzchołek 2:

- Kolejność gdy przeszukujemy wszerz: 1, 2,
- Kolejność gdy przeszukujemy w głąb: 1, 2,

