1. Zadania rozwiązywane bez użycia komputera 5

# Zadania rozwiązywane bez użycia komputera

**1.1. Analiza algorytmów**

**Zadanie 1.**

**Wiązka zadań *Ciągi rekurencyjne***

Dana jest następująca funkcja rekurencyjna:

**funkcja *wynik*( *i* )**

**jeżeli** *i* < 3

**zwróć** 1 i **zakończ**;

**w przeciwnym razie**

**jeżeli** *i* mod 2 = 0

**zwróć** *wynik*(*i* – 3) + *wynik*(*i* – 1) + 1

**w przeciwnym razie**

**zwróć** *wynik*(*i* – 1) mod 7

**Uwaga:** Operator mod oznacza resztę z dzielenia.

**1.1.**

Uzupełnij poniższą tabelę:

|  |  |
| --- | --- |
| ***i*** | **wynik(*i*)** |
| 2 | 1 |
| 3 | 1 |
| 4 | 3 |
| 5 | 3 |
| 6 | 5 |
| 7 | 5 |
| 8 | 9 |

**1.2.**

*Wykonaniem elementarnym* nazywać będziemy wykonanie *wynik*(0), *wynik*(1) lub *wynik*(2). Natomiast *złożonością elementarną* *wynik*(*i*) nazywamy liczbę *wykonań elementarnych* będących efektem uruchomienia *wynik*(*i*). Złożoność elementarną *wynik*(*i*) oznaczamy przez *E*(*i*).

Na przykład złożoność elementarna *wynik*(4) wynosi *E*(4) = 2, ponieważ wykonując *wynik*(4), wywołamy *wynik*(3) i *wynik*(1) (wykonanie elementarne), a z kolei przy wykonaniu *wynik*(3) wywołamy *wynik*(2) (drugie wykonanie elementarne).

Uzupełnij poniższą tabelę:

6 Egzamin maturalny. Informatyka. Poziom rozszerzony. Zbiór zadań

|  |  |
| --- | --- |
| ***i*** | ***E*(*i*)** |
| 0 | 1 |
| 3 | 1 |
| 5 | 2 |
| 7 | 3 |
| 9 | 5 |
| 10 | 8 |

Okazuje się, że *E*(*i*) można opisać rekurencyjnym wyrażeniem, którego niekompletną postać podajemy poniżej. Uzupełnij brakujące miejsca tak, aby *E*(*i*) dawało poprawną złożoność elementarną *wynik*(*i*) dla każdego całkowitego nieujemnego *i*.

E(0) = E(1) = E(2) = 1

E(*i*) = E(i - 3) + E(i - 1) dla parzystego *i* > 2

E(*i*) = E(i - 1) dla nieparzystego *i* > 2

**1.3.**

Naszym celem jest wyznaczenie największej liczby spośród wartości funkcji *wynik*(0), *wynik*(1),…,*wynik*(1000) bez konieczności rekurencyjnego wyznaczania kolejnych wartości. Poniżej prezentujemy niekompletny algorytm realizujący to zadanie.

*W*[0] ← 1

*W*[1] ← 1

*W*[2] ← 1 *max\_wart* ← 1

**dla** *i* = 3, 4, …, 1 000 **wykonuj**  **jeżeli** *i* mod 2 = 0

*W*[*i*] ← W[*i – 3*] + W[*i – 1*] + 1

**w przeciwnym razie**

*W*[*i*] ← W[*i – 1*] mod 7

**jeżeli** *W*[*i*] > *max\_wart*

*max\_wart* ← *W[i]*

**zwróć** *max\_wart*

Uzupełnij brakujące miejsca w algorytmie tak, aby zwracał on największą liczbę spośród *wynik*(0), *wynik*(1),…,*wynik*(1000).

**Komentarz do zadania**

**1.1.**

Do rozwiązania tego zadania stosujemy definicję rekurencyjną *wynik*(*i*), wynikającą wprost z podanego pseudokodu:

*wynik*(0) = *wynik*(1) = *wynik*(2)=1 *wynik*(*i*) = *wynik*(*i* – 3) + *wynik*(*i* – 1) + 1 dla parzystych *i*>2 *wynik*(*i*) = *wynik*(*i* – 1) mod 7 dla nieparzystych *i*>2 A zatem:

* *wynik*(3) = *wynik*(2) mod 7 = 1 mod 7 = 1
* *wynik*(4) = *wynik*(3) + *wynik*(1) + 1 = 1 + 1 + 1 = 3