1	2	3	4	5	6	7	8	9	10	Σ
-	-	-	+	-	+	+	+	-	+	???

ZADANIE 4.

Udowodnij, że algorytm mnożenia liczb "po rosyjsku" jest poprawny. Jaka jest jego złożoność czasowa i pamięciowa przy:

1. jednorodnym kryterium kosztów,

2. logarytmicznym kryterium kostów?

Dowodzik:

Ustalmy dowolne $b \in \mathbb{N}$. Pokażemy przez indukcję, że dla dowolnego $a \in \mathbb{N}$ wynik algorytmu jest równy ab. Przypadek bazowy, czyli $a = 1$ jest trywialny.

Załóżmy teraz, że dla dowolnego $a' \leq n$ algorytm działa i niech $a = n + 1$. Rozważmy dwa przypadki:

1. $(n + 1)$ jest nieparzyste. Wtedy $a_1 = (n + 1)$ jest nieparzyste, wpp. do $a'_1 = n$. Dalej, $a_2 = \lfloor \frac{n+1}{2} \rfloor = \lfloor \frac{n}{2} \rfloor = a'_2$, czyli od drugiego miejsca ciąg a_i dla $n + 1$ jest taki sam jak dla n, więc:

$$\sum_{i=1,a_i\ np}^{k} b_i = b_1 + \sum_{i=2,a'_i\ np} b_i = b_1 + n \cdot b = b + bn = b(n + 1)$$

2. $(n + 1)$ jest parzyste. Wtedy $a_1 = (n + 1)$ jest parzyste, więc b_1 nie zostanie użyte w sumie. Natomiast $\lfloor \frac{n+1}{2} \rfloor = \lfloor \frac{n}{2} \rfloor + 1 = a'_2 + 1$, czyli od trzeciego indeksu a_i jest taki sam jak a'_i

$$\sum_{i=1,a_i\ np}^{k} b_i = a_2 b_2 + \sum_{i=3,a_i\ np}^{k} b_i = (a'_2 + 1)2b + \sum_{i=3}^{k} b_i = 2b + \sum_{i=2,a'_i\ np}^{k} b_i = 2b + \sum_{i=1}^{k} b_i - b_1 = b + \sum_{i=1}^{k} b_i = b + nb$$

Złożoność:

1. Będziemy obliczać wyrazy ciągu a_i $\log_2(a)$ razy, za każdym razem będziemy dzielić, sprawdzać podzielność, mnożyć b i ewentualnie dodawać, co jest mniej więcej stałą liczbą operacji.

2. ?????

ZADANIE 5.

Oszacuj z dokładnością do Θ złożoność poniższego fragmentu programu:

```
res <- 0
for i <- 1 to n do
    j <- i
    while (j jest parzyste) j <- j/2
    res <- res + j
```

ZADANIE 6.

Pokaż, w jaki sposób algorytm "macierzowy" obliczania n-tej liczby Fibonacciego można uogólnić na inne ciągi, w których kolejne elementy definiowane są liniową kombinacją skończonej liczby elementów wcześniejszych. Następnie uogólnij swoje rozwiązanie na przypadek, w którym n-ty element ciągu definiowany jest jako suma kombinacji liniowej skończonej liczby elementów wcześniejszych oraz wielomianu zmiennej n.

Pierwsza część jest dość prosta. Rozważamy ciąg zdefiniowany rekurencyjnie

$$a_n = \alpha_1 a_{n-1} + \alpha_2 a_{n-2} + \ldots + \alpha_k a_{n-k}.$$

Popatrzmy na macierz

$$A = \begin{bmatrix} \alpha_1 & \alpha_2 & \alpha_3 & \ldots & \alpha_{k-1} & \alpha_k \\ 1 & 0 & 0 & \ldots & 0 & 0 \\ 0 & 1 & 0 & \ldots & 0 & 0 \\ \ldots & \ldots & \ldots & \ldots & \ldots & \ldots \\ 0 & 0 & 0 & \ldots & 1 & 0 \end{bmatrix}$$

Mnożąc A^{n-1} przez wektor

$$\begin{bmatrix} a_{k-1} \\ a_{k-2} \\ \dots \\ a_0 \end{bmatrix}$$

dostajemy wektor zawierający a_n na pierwszym miejscu oraz wszystkie poprzednie miejsca na pozostałych miejscach.

Teraz co się dzieje dla ciągu zawierającego wielomian zmiennej n?

Przyjrzyjmy się ciągowi zdefiniowanemu rekurencyjnie ze szczyptą wielomianu:

$$a_n = \sum_{i=1}^{k} \alpha_i a_{n-i} + \sum_{i=0}^{m} \beta_i n^i$$