4. Pokażmy, że algorytm mnożenia „po rosyjsku" jest poprawny.

Weźmy dowolne a, b i wyznaczmy $a \cdot b$. Wiemy, że a można przedstawić jako $a = \sum_{i=0}^{n} 2^i$ (i-ta cyfra w zapisie binarnym a to 1), np. $21_{10} = 10101_2 = 2^4 + 2^2 + 2^0$. Ogólnie $a = 2^{k_0} + 2^{k_1} + 2^{k_2} + \ldots + 2^{k_n}$.

Wtedy $a \cdot b = (2^{k_0} + 2^{k_1} + \ldots + 2^{k_n}) \cdot b = 2^{k_0} \cdot b + 2^{k_1} \cdot b + \ldots + 2^{k_n} \cdot b$.

Zauważmy, że algorytm mnożenia po rosyjsku wyznaczając kolejne $a_{i+1} = \lfloor \frac{a_i}{2} \rfloor$ wyznacza $a_i \gg 1$, tzn. przesuwa cyfry w zapisie binarnym a o jedno miejsce w prawo. Wiemy, że liczba jest parzysta, gdy jej najmniej znacząca cyfra w zapisie binarnym to 0. Zauważmy także, że obliczając kolejne a_i mamy wyznaczoną rozkładkę a na potęgi dwójek, tzn. jeśli a_i jest nieparzyste, czyli jego najmniej znacząca cyfra to 1, wtedy i-ta cyfra w zapisie binarnym a to 1, czyli 2^i występuje w rozkładzie a na potęgi dwójek. Algorytm do sumy wynikowej bierze tylko te wyrazy $2^i \cdot b$, gdzie a_i było nieparzyste, tzn. otrzymamy sumę $\sum_{i=1}^{k} b_i = \sum_{i=1}^{k} 2^{i-1} \cdot b =$ (a_i np. / a_i n.p.)

$= 2^{k_1} b + 2^{k_2} b + \ldots + 2^{k_n} b$, gdzie $k_1, \ldots, k_n$ to wartości i, dla których a_i jest nieparzyste. Stąd

Wiemy, że $\sum_{i=1}^{k} b_i = (2^{k_1} + 2^{k_2} + \ldots + 2^{k_n}) b = a \cdot b$, zatem algorytm jest poprawny.

(a_i np.)

Złożoność przy jednorodnym kryterium kosztów dla obliczania $a \cdot b$:

- czasowa: $O(\log a)$, ponieważ wykonujemy operacje dzielenia a przez 2 tyle razy, ile razy możemy a podzielić przez 2 i w każdej iteracji dzielimy a przez 2, mnożymy b przez 2, co między wykonaniem przesunięciem bitowym i dodajemy b do sumy wynikowej, jeśli a_i jest nieparzyste (można sprawdzić $b \,\&\, 1 == 1$), stąd mamy $4 \cdot \log(a)$ operacji.
- pamięciowa: $O(1)$, bo potrzebujemy tylko zmiennej a, b i wyniku, których wartości zmieniamy w każdej iteracji.

Złożoność przy logarytmicznym kryterium kosztów:

- czasowa: $\log(a) \cdot (\log(a) + \log(b)) \cdot 2 = O(\log(a) \cdot \log(ab))$
- pamięciowa: $\log(a) + \log(b) + \log(a+b) = \log(ab) + \log(a+b) = \log(ab(a+b)) =$

$= O(\log(a^2 b + ab^2))$