Zadanie 3.6. Egzamin maj 2011 r. Arkusz I, poziom rozszerzony, zadanie 1. DŁUGOŚĆ NAPISÓW BINARNYCH

Opisana poniżej funkcja rekurencyjna wyznacza, dla liczby naturalnej $n > 0$, długość napisu uzyskanego przez sklejenie binarnych reprezentacji liczb naturalnych od 1 do $n - 1$.

Funkcja sklej (n)

krok 1. jeśli $n = 1$, to podaj 0 jako wynik i zakończ działanie

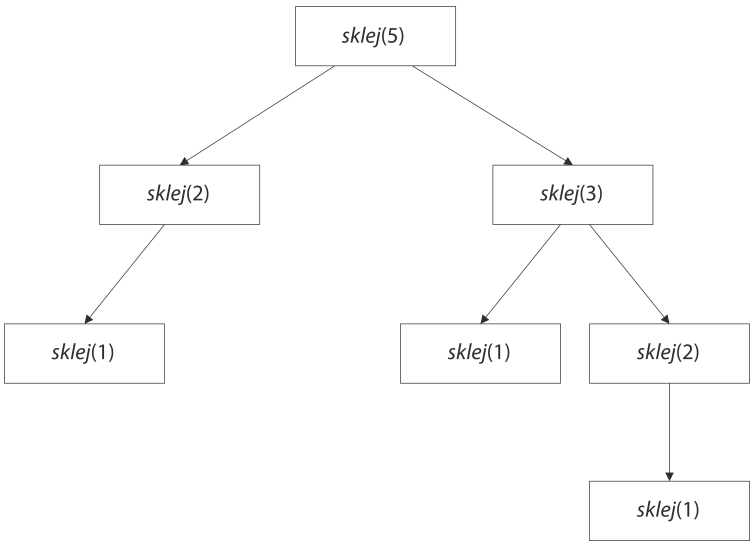
krok 2. jeśli n parzysta, to wynikiem jest $n - 1 + 2 \times sklej\ (n / 2)$

krok 3. jeśli n nieparzysta, to wynikiem jest $n - 1 + sklej\ ((n - 1) / 2) + sklej\ ((n + 1) / 2)$

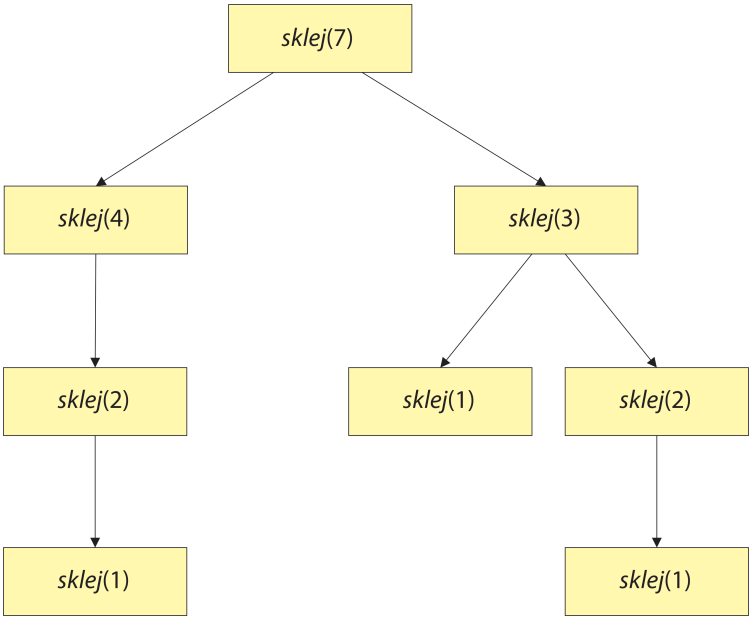
Wykonaj polecenia a)–c):

a) Wykonanie funkcji sklej można przedstawić w postaci drzewa wywołań rekurencyjnych ilustrującego wszystkie wywołania funkcji po jej uruchomieniu dla zadanego argumentu.

Poniższy rysunek przedstawia takie drzewo dla wywołania *sklej*(5).



Narysuj analogiczne drzewo dla wywołania *sklej*(7).



b) Uzupełnij poniższą tabelę, podając wartości funkcji *sklej* dla wskazanych argumentów.

n	*sklej(n)*
1	0
2	1
3	3
4	5
5	8
6	11

Chcemy wypełnić tablicę $s[1..n]$ w taki sposób, że $s[i] = sklej(i)$ dla każdego $1 \le i \le n$.

Podaj algorytm wypełniający tablicę s odpowiednimi wartościami **bez wywoływania funkcji *sklej***, tzn. **bez** użycia **rekurencji**. Zauważ, że jeśli poprawnie wyliczone są już wartości $s[1]$, ..., $s[i-1]$, to można z nich skorzystać przy wyznaczaniu $s[i]$.

Zapisz swój algorytm w postaci listy kroków, schematu blokowego lub w wybranym języku programowania, który wybrałeś/aś na egzamin.

Specyfikacja:

Dane: liczba naturalna $n > 0$

Wynik: tablica $s[1..n]$ o wartościach $s[i] = sklej\ (i)$, dla $1 \le i \le n$

***Listing** (zad_c.py):*

```
# rekurencyjnie
def sklej(n):
    if n == 1: return 0
    if n % 2 == 0: return n - 1 + 2 * sklej(n // 2)
    else: return n - 1 + sklej((n - 1) // 2) + sklej((n + 1) // 2)

# iteracyjnie
def zad_c(n):
    s = [0 for row in range(n + 1)]
    for i in range(2, n + 1):
        if i % 2==0: s[i] = i - 1 + 2 * s[i // 2]
        else: s[i] = i - 1 + s[(i - 1) // 2] + s[(i + 1) // 2]
    return s

print(sklej(10))
print(zad_c(10))
```