

Zadanie 1 (2p.)

Napisz funkcję znajdującą wszystkie podlisty (rozumiane jako podciągi) listy zadanej jako argument. Czy umiesz napisać tę funkcję tak by nie generowała nieużytków?

Zadanie 2 (2p.)

Napisz dwie wersje funkcji (w tym jedną za pomocą rekursji ogonowej), która oblicza n-ty wyraz ciągu zdefiniowanego wzorem:

- $a_0 = 1$
- $a_1 = 2$
- $a_n = 2 * a_{n-2} - a_{n-1} + 1$ dla $n > 1$

a następnie porównaj ich działanie dla dużych n.

Zadanie 3 (2p.)

Napisz dwie wersje funkcji (w tym jedną za pomocą rekursji ogonowej), która dla zadanej listy l oraz funkcji f zwraca listę będącą odwróceniem listy l przetworzonej przez działanie funkcji f. Każdy element listy może być odwiedzony tylko raz.

Zadanie 4 (3p.)

1. Napisz funkcję merge, która łączy dwie listy posortowane rosnąco w pewnym porządku \leq tak, by wynik działania funkcji był także listą posortowaną rosnąco w tym samym porządku. Argumentami funkcji merge powinny być: funkcja cmp typu 'a -> 'a -> bool (zakładamy, że $\text{cmp } a \text{ } b = \text{true}$ wtw $a \leq b$), i dwie listy elementów typu 'a. Przykład: $\text{merge } (\leq) [1;2;5] [3;4;5]$ utworzy listę $[1;2;3;4;5;5]$.
2. Zapisz tę funkcję używając rekursji ogonowej, a następnie porównaj działanie obu funkcji na odpowiednich przykładach.
3. Wykorzystaj funkcję merge do napisania funkcji sortowania przez scalanie.

Zadanie 5 (2p.)

Napisz funkcję zwracającą listę wszystkich permutacji zadanej listy.

Zadanie 6 (2p.)

Napisz funkcję generującą wszystkie sufiksy danej listy. Na przykład dla listy [1;2;3] Twoja funkcja powinna zwrócić listę [[1; 2; 3]; [2; 3]; [3]]. Następnie, napisz funkcję generującą wszystkie prefiksy danej listy. Na przykład dla listy [1; 2; 3] Twoja funkcja powinna zwrócić listę [[1]; [1; 2]; [1; 2; 3]].