

Lista 3

Zadanie 1 (2p.)

Założmy, że wielomiany o współczynnikach rzeczywistych są reprezentowane jako listy współczynników od najwyższej potęgi do najniższej, np. $[1.;0.;-1.;2.]$ oznacza wielomian $x^3 - x + 2$. Napisz funkcję, która dla zadanej listy reprezentującej wielomian i dla danego argumentu x typu float, obliczy wartość tego wielomianu w punkcie x w oparciu o schemat Hornera. Napisz tę funkcję w dwóch wersjach: raz za pomocą rekursji ogonowej, a następnie bez jawnego użycia rekursji, korzystając z odpowiedniej funkcji bibliotecznej modułu List.

Zadanie 2 (2p.)

Założmy, że zmieniamy reprezentację wielomianu tak, by współczynniki w liście były uszeregowane od najniższej potęgi do najwyższej (wtedy $[1.;0.;-1.;2.]$ oznacza wielomian $2x^3 - x^2 + 1$). Zmodyfikuj obie funkcje z Zadania 1 tak, by poprawnie liczyły wartość wielomianu w tej reprezentacji (rekursja w rozwiązaniu z jawną rekursją nie musi być ogonowa).

Zadanie 3 (2p.)

Dla reprezentacji wielomianu z Zadania 2 napisz funkcję (ponownie w dwóch wersjach) obliczającą pochodną wielomianu (np. dla listy $[1.;0.;-1.;2.]$ funkcja ta powinna utworzyć listę $[0.;-2.;6.]$).

Zadanie 4 (6p.)

Niech macierz kwadratowa $n \times n$ będzie reprezentowana wierszami jako lista list.

1. Napisz funkcję sprawdzającą, czy dana lista jest poprawną reprezentacją macierzy kwadratowej.
2. Napisz funkcję, która dla zadanej macierzy kwadratowej i liczby naturalnej n wyznacza n -tą kolumnę macierzy.
3. Wykorzystaj funkcję z poprzedniego punktu do napisania funkcji transpozycji macierzy, np. transpozycja macierzy l jest reprezentowana jako lista $[[1.;4.;7.]; [2.;5.;8.]; [3.;6.;9.]]$.
4. Napisz funkcję zip, która dla danych dwóch list równej długości tworzy listę złożoną z par elementów obu list znajdujących się na tych samych pozycjach, np. $\text{zip } [1.;2.;3.] ["a";"b";"c"] = [(1., "a"); (2., "b"); (3., "c")]$.
5. Korzystając z funkcji zip, napisz funkcję zipf, która dla danych dwóch list typów 'a' list i 'b' list i funkcji dwuargumentowej f typu 'a -> 'b -> 'c' tworzy listę złożoną z

wartości funkcji f na argumentach z obu list położonych na tych samych pozycjach, np. `zipf (+.) [1.;2.;3.] [4.;5.;6.] = [5.;7.;9.]`.

6. Wykorzystując funkcję `zipf` napisz funkcję `mult_vec`, która oblicza iloczyn zadanego wektora i zadanej macierzy, np. `mult_vec [1.;2.] [[2.;0.];[4.;5.]] = [10.;10.]`.
7. Korzystając z funkcji `mult_vec` napisz funkcję mnożenia dwóch macierzy kwadratowych tego samego rozmiaru.

Uwaga: Wykorzystaj funkcje biblioteczne operujące na listach tak by Twoje rozwiązania były zwięzłe.

Zadanie 5 (4p.)

Napisz funkcję, która dla zadanej permutacji elementów dowolnego typu 'a', na którym zdefiniowany jest pewien porządek liniowy, znajduje kolejną - w porządku leksykograficznym - permutację tych samych elementów, np. dla permutacji (1,2,4,3) funkcja powinna zwrócić permutację (1,3,2,4), a dla permutacji (a,c,b) wynikiem powinna być permutacja (b,a,c). W przypadku, w którym zadana permutacja jest największa, funkcja powinna zwracać permutację najmniejszą. Do reprezentowania permutacji użyj list, ale w porządku odwróconym, tj. w przykładzie powyżej funkcja dla argumentu [3;4;2;1] powinna dać odpowiedź [4;2;3;1], a dla ['b';'c';'a'] - odpowiedź ['c';'a';'b']. Wykorzystaj tę funkcję do napisania funkcji generującej wszystkie permutacje danej listy.

Zadanie 6 (2p.)

Pewnego razu powiedziano dwóm logikom P i S, że wybrano dwie liczby naturalne x i y takie, że $1 < x < y$ oraz $x+y < 100$. Co więcej, S otrzymał informację o wartości $x+y$, a P o wartości $x*y$. Po chwili P i S odbyli następującą rozmowę:

P: Nie potrafię powiedzieć jakie to liczby.

S: Wiedziałem o tym.

P: A to w takim razie już potrafię.

S: Ja już też.

Napisz program, który wykorzysta powyższy dialog do znalezienia liczb x i y .