

Języki i paradygmaty programowania

Lista nr 5 na laboratorium 12 i 13

Zadanie 1

Stwórz w Haskellu, Lispie i SMLu programy obliczające następujące funkcje:

1. `binomial n k` obliczający dwumian Newtona $\binom{n}{k}$ z wzoru rekurencyjnego $\binom{n}{k} = \binom{n-1}{k} + \binom{n-1}{k-1}$ i warunków brzegowych $\binom{n}{0} = \binom{n}{n} = 1$;
2. `binomial2 n k` wykorzystujący do obliczeń kolejne wiersze trójkąta Pascala (w postaci list) i zwracająca k -ty element n -tego wiersza (*porównaj szybkość i zasięg obu rozwiązań*);
3. `mergesort L` implementującą sortowanie przez scalanie;
4. `de a b` zwracającą rozwiązanie równania diofantycznego $ax + by = z = \gcd(a, b)$ na liczbach całkowitych, w postaci krotki (x, y, z) (*dla Lisp-a w postaci listy*);
5. `prime_factors n` zwracającą listę rozkładu n na czynniki pierwsze;
6. `totient n` zwracającą wartość funkcji Eulera dla n napisaną za pomocą klasycznego algorytmu sprawdzającego które liczby od 1 do n są względnie pierwsze z n ;
7. `totient2 n` zwracającą wartość funkcji Eulera dla n policzoną na podstawie wyniku funkcji `prime_factors n` (*porównaj szybkość i zakres działania obu funkcji liczących funkcję Eulera*);
8. `primes n` zwracającą listę wszystkich liczb pierwszych między 2 a n .

(W Lispie proszę nie używać funkcji `loop`)

Zadanie 2

Stwórz w Prologu predykaty obliczające następujące funkcje:

1. `mergesort(+X,-Y)` implementujący sortowanie przez scalanie listy X do listy Y ;
2. `de(+A,+B,-X,-Y,-Z)` zwracający rozwiązanie równania diofantycznego $ax + by = z = \gcd(a, b)$ na liczbach całkowitych;
3. `prime_factors(+N,-X)` zwracający listę X rozkładu N na czynniki pierwsze;
4. `totient(+N,-T)` zwracający wartość funkcji Eulera T dla N ;
5. `primes(+N,-X)` zwracającą listę X wszystkich liczb pierwszych między 2 a N .