

Programowanie funkcyjne – laboratorium 2

14.10.2009 r.

Standardowe funkcje na listach

1. `head [] = Exception: Failure "nth"`
`head [1;2;3] = 1`
2. `tail [] = []`
`tail [1;2;3] = [2;3]`
3. `from_to 1 5 = [1;2;3;4;5]`
`from_to 7 3 = []`
4. `nth 0 ["a";"b";"c";"d"] = "a"`
`nth 2 ["a";"b";"c"] = "c"`
`nth 5 ["a"] = nth (-2) ["a"] = Exception: Failure "nth"`
5. `last [1;2;3] = 3`
`last [] = Exception: Failure "last"`
6. `take 3 [1;2;3;4;5] = [1;2;3]`
`take 7 [1] = [1]`
7. `drop 3 [1;2;3;4;5] = [4;5]`
`drop 7 [1] = []`
8. `app ['a';'b'] ['c';'d';'e'] = ['a';'b';'c';'d';'e']`
`app [] ['a';'b'] = app ['a';'b'] [] = ['a';'b']`
9. `rev [2;5;8] = [8;5;2]`
10. `interleave [1;3;5;7;9] [2;4;6] = [1;2;3;4;5;6]`
11. `zip [1;2;3;4;5] ["ala";"ma";"kota"] = [(1,"ala"); (2,"ma"); (3,"kota")]`
12. `unzip [(1,"ala"); (2,"ma"); (3,"kota")] = ([1;2;3], ["ala";"ma";"kota"])`
13. `flatten [[1;2];[];[3;4;5]] = [1;2;3;4;5]`
14. `iter 0 f 5 = [5]`
`iter 3 f 5 = [5; f 5; f (f 5); f (f (f 5))];;`

Niestandardowe funkcje na listach liczb

1. Lista różnicowa listy $[x_1; \dots; x_n]$ to lista różnic: $[(x_1 - x_2); (x_2 - x_3); \dots; (x_{n-1} - x_n)]$ n-ta lista różnicowa to lista różnicowa (n-1)-szej listy różnicowej (pierwsza lista różnicowa to lista różnicowa).
`diff_list 1 [5;3;2;0] = [2;1;2]`
`diff_list 2 [5;3;2;0] = diff_list 1 [2;1;2] = [1;-1]`
2. `max_diff [x_1; ... x_n] = max { x_k - x_l | 0 < k,l <= n }`
`max_diff [] = 0`
3. Jeśli l jest listą liczb dodatnich, uporządkowanych rosnąco, to `(triples l)` jest listą trójek (a,b,c) , takich, że a,b,c należą do l , $a < b < c$ oraz $c < a + b$
4. `max_sum [x_1; ... x_n] = (k,l)`, t.ż. $0 < k,l <= n$ oraz $x_k + \dots + x_l = \max \{ x_p + \dots + x_q \mid 0 < p,q <= n \}$

Sortowanie list

1. mergesort
2. quicksort