ГУАП

КАФЕДРА № 43

ОТЧЕТ   
ЗАЩИЩЕН С ОЦЕНКОЙ

ПРЕПОДАВАТЕЛЬ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| доцент |  |  |  | Степанов П.А. |
| должность, уч.степень, звание |  | подпись, дата |  | инициалы, фамилия |

|  |
| --- |
| ОТЧЕТ О ЛАБОРАТОРНОЙ РАБОТЕ №3 |
| Программирование алгоритмов перебора |
| по курсу: ФУНКЦИОНАЛЬНОЕ И ЛОГИЧЕСКОЕ ПРОГРАММИРОВАНИЕ |
|  |
|  |

РАБОТУ ВЫПОЛНИЛ

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| СТУДЕНТ ГР. № | 4136 |  | 09.06.2023 |  | Н. С. Бобрович |
|  |  |  | подпись, дата |  | инициалы, фамилия |

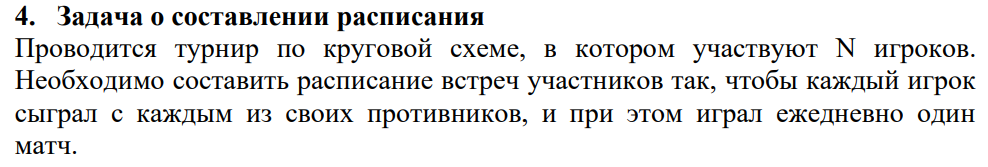
Санкт-Петербург 2023

**Цель:** исследование класса задач, решаемых полным перебором и методов их решения средствами Lisp.

**Указания:**

Используя списковое представление графов и рекурсивные функции, разработать программу, находящую решение, в соответствии с приведенными ниже вариантами. Не разрешается использовать операторы цикла и оператор PROG.

**Задание:**

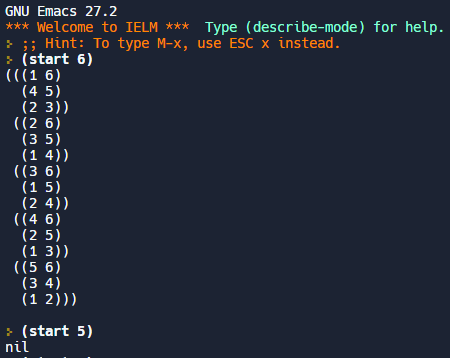


**Ход работы.**

**Листинг:**

(require 'cl-lib)  
  
(defun get-set-games (n)   
 (cond   
 ((<= n 1) nil)  
 (t (append (combine-n (- n 1) n) (get-set-games (- n 1))))  
 )  
)  
  
(defun combine-n (n m)   
 (cond   
 ((<= n 0) nil)  
 (t (append (list (list n m)) (combine-n (- n 1) m)))  
 )  
)  
  
(defun get-set-players (n)   
 (cond   
 ((<= n 0) nil)  
 (t (cons n (get-set-players (- n 1))))  
 )  
)  
  
(defun delete-list1-from-list2 (list1 list2)   
 (set-difference list2 list1 :test #'equal)  
)  
  
(defun get-day (set-players set-games)   
 (cond   
 ((null set-players) nil)  
 ((null set-games) nil)  
 ((subsetp (car set-games) set-players)  
 (cons (car set-games)   
 (get-day   
 (delete-list1-from-list2 (car set-games) set-players)  
 (cdr set-games)  
 )  
 )  
 )  
 (t (get-day set-players (cdr set-games)))  
 )  
)  
  
(defun get-valid-day (set-players set-games &optional day)   
 (cond   
 ((null day)  
 (get-valid-day set-players set-games (get-day set-players set-games))  
 )  
 ((equal (length day) (/ (length set-players) 2)) day)  
 (t  
 (setq new-set-players (delete-list1-from-list2 (car day) set-players))  
 (setq new-set-games (delete-list1-from-list2 day set-games))  
 (cons   
 (car day)  
 (get-valid-day   
 new-set-players  
 new-set-games  
 (get-day   
 new-set-players  
 new-set-games  
 )  
 )  
 )  
 )  
 )  
)  
  
(defun get-schedule (set-players set-games &optional result)   
 (cond   
 ((null set-games) result)  
 (t  
 (setq tmp (get-valid-day set-players set-games))  
 (get-schedule   
 set-players  
 (delete-list1-from-list2 tmp set-games)  
 (cons tmp result)  
 )  
 )  
 )  
)  
  
(defun start (n)   
 (cond   
 ((< n 0) nil)  
 ((equal (mod n 2) 1) nil)  
 (t  
 (progn  
 (setq set-players (get-set-players n))   
 (setq set-games (get-set-games n))  
 (get-schedule set-players set-games))  
 )  
 )  
)

**Output:**



**Вывод:** Исследовал класс задач, решаемых полным перебором и методов их решения средствами Lisp.