
КАФЕДРА №

ОТЧЕТ
ЗАЩИЩЕН С ОЦЕНКОЙ
ПРЕПОДАВАТЕЛЬ

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. №

подпись, дата

инициалы, фамилия

Санкт-Петербург 2023

Цель: Исследование класса задач, решаемых полным перебором и методов их решения средствами Lisp

Задание на лабораторную работу:

Вариант 14. Вывести в произвольном орграфе все имеющиеся циклы без дублирования

Листинг программы

```
(SETQ GRAPH '((A B C) (B A) (C D) (D A)))
;; (SETQ GRAPH '((A B C) (B D) (D A)))

; вход: граф
; выход: список точек
(DEFUN POINTS (org)
  (COND
    ((NULL org) ())
    (T (CONS (CAAR org) (POINTS (CDR org)))))
  )
)

; вход: список
;       символ
; выход: обрезанный список с конца по символ
(DEFUN CUT_LAST_ON_SYMBOL (lst symbol)
  (COND
    ((NULL lst) NIL)
    ((EQ (CAR (LAST lst)) symbol) lst)
    (T (CUT_LAST_ON_SYMBOL (BUTLAST lst) symbol)))
  )
)

; вход: список
; выход: обрезанный список, у которого
;        первый и последний элементы одинаковые
(DEFUN CUT_ON_CYCLE (lst)
  (COND
    ((NULL lst) NIL)
    ((FIND_IN_LIST (CDR lst) (CAR lst))
     (CUT_LAST_ON_SYMBOL lst (CAR lst)))
    (T (CUT_ON_CYCLE (CDR lst)))
  )
)

; вход: список
;       значение
; выход: Найден или не найден символ (T или NIL)
(DEFUN FIND_IN_LIST (lst value)
  (COND
    ((NULL lst) NIL)
    ((eq (CAR lst) value) T)
    (T (FIND_IN_LIST (cdr lst) value))
  )
)
```

```

)
)

; вход: список
;      значение
; выход: список без найденного значения
(DEFUN REMOVE_ELEMENT(lst elem)
  (COND
    ((NULL lst) NIL)
    ((EQUAL (CAR lst) elem) (REMOVE_ELEMENT (CDR lst) elem))
    (T (CONS (CAR lst) (REMOVE_ELEMENT (CDR lst) elem)))
  )
)

; вход: список
;      символ
;      счетчик переборов
; выход: список, в котором выведены все возможные
;      варианты повторения цикла
(DEFUN VARIANTS(lst elem len)
  (COND
    ((<= len 2) lst)
    (T (VARIANTS (REMOVE_ELEMENT lst elem) (APPEND (CDR elem) (LIST (CADR elem))) (- len 1)))
  )
)

; вход: список
; выход: список, в котором удалены все повторяющиеся циклы
(DEFUN DEL_DUPLICATE(lst)
  (COND
    ((NULL lst) NIL)
    (T (CONS (CAR lst) (DEL_DUPLICATE
      (CDR (VARIANTS lst (CAR lst) (LENGTH (CAR lst)))))))
  )
)

; вход: граф
; выход: список циклов
(DEFUN SEARCH_CYCLES_GRAPH(GRAPH)
  (COND
    ((NULL GRAPH) NIL)
    (T (DEL_DUPLICATE (DEFI NIL GRAPH NIL (LIST (CAR (POINTS graph))) (POINTS graph))))
  )
)

; вход: уже найденные циклы
;      граф
;      список посещённых вершин
;      путь по которому мы прошли
;      список начальных посещенных вершин
; выход: список всех найденных циклов
(DEFUN DEFI (RES GRAPH VISITED PATH VISITED_ROOT)

```



```

)

; вход: вершина
;      граф
; выход: список вершин в которые можно перейти
(DEFUN NEIGHBOUR3 (X GRAPH)
  (COND
    ((NULL (ASSOC X GRAPH)) NIL)
    (T (CDR (ASSOC X GRAPH))))
  )
)

; вход: граф
; выход: список всех найденных циклов
(DEFUN SEARCH_CYCLES_GRAPH(GRAPH)
  (COND
    ((NULL GRAPH) NIL)
    (T (DEL_DUPLICATE (DEFI NIL GRAPH NIL (LIST (CAR (POINTS GRAPH))) (POINTS GRAPH))))
  )
)

(write graph)(terpri)
(write (SEARCH_CYCLES_GRAPH graph))

```

Пример выполнения программы:

```

((A B C) (B A) (C D) (D A))
((A B A) (C D A C))
[Done] exited with code=0 in 0.114 seconds

```

Вывод:

Проведено исследование класса задач, решаемых полным перебором и методов их решения средствами Lisp