Отчет по лабораторной работе №3

По курсу Логическое программирование.

Работу выполнил

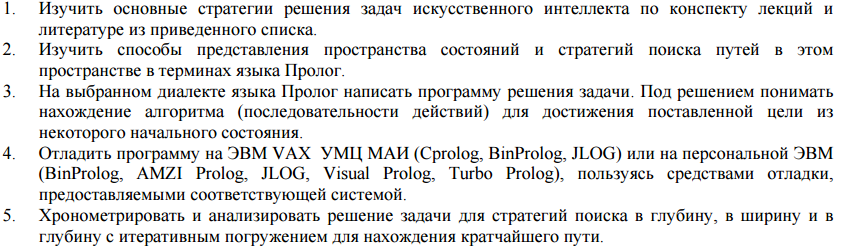
студент группы 8О-207Б

Сорокин Денис Михайлович

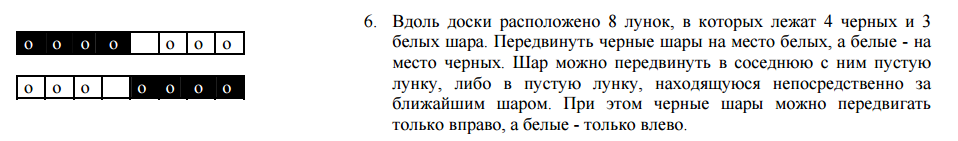
Сдано: \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

**Тема:**

Применение Пролога для решения задач поиска в пространстве состояний.

**Цель работы:**

**Задание:**



**Текст программы:**

%Программа написана и выполнена в web версии SWI-Prolog – SWISH http://swish.swi-prolog.org/

%member

**member**(X, [X|**\_**]).

**member**(X, [**\_**|Tail]) :-

**member**(X, Tail).

%append

**append**([], X, X).

**append**([X|T], Y, [X|A]) :-

**append**(T, Y, A).

**step**(A,B):-

**append**(Begin,["\_","w"|Tail],A),

**append**(Begin,["w","\_"|Tail],B).

**step**(A,B):-

**append**(Begin,["b","\_"|Tail],A),

**append**(Begin,["\_","b"|Tail],B).

**step**(A,B):-

**append**(Begin,["\_","b","w"|Tail],A),

**append**(Begin,["w","b","\_"|Tail],B).

**step**(A,B):-

**append**(Begin,["b","w","\_"|Tail],A),

**append**(Begin,["\_","w","b"|Tail],B).

**print\_ans**([]).

**print\_ans**([H|Tail]):-

**print\_ans**(Tail),

**write**(H), nl.

**dFS**(Goal,[Goal|Tail]) :-

!, **print\_ans**([Goal|Tail]).

**dFS**(Goal, [Curr|Tail]) :-

**step**(Curr, Tmp),

not(**member**(Tmp,Tail)),

**dFS**(Goal, [Tmp, Curr|Tail]).

**next**(X, HasBeen, Y) :-

**step**(X,Y),

not(**member**(Y,HasBeen)).

**bFS**([First|**\_**],Goal,First) :-

First = [Goal|**\_**].

**bFS**([[LastWay|HasBeen]|OtherWays],Finish,Way):-

**findall**([Z,LastWay|HasBeen],

**next**(LastWay, HasBeen, Z), List),

**append**(List,OtherWays,NewWays),

**bFS**(NewWays,Finish,Way).

**prolong**([X|T],[Y,X|T]) :-

**step**(X,Y),

not(**member**(Y,[X|T])).

**switer**([Goal|Tail],Goal,[Goal|Tail],**0**).

**switer**(TempWay,Goal,Way,N):-

N > **0**,

**prolong**(TempWay,NewWay),

N1 is N-**1**,

**switer**(NewWay,Goal,Way,N1).

**for**(X, X, **\_**).

**for**(I, X, Y):-

X < Y,

X1 is X + **1**,

**for**(I, X1 ,Y).

**solve**(Start, Goal):-

**write**('ПОИСК С ИТЕРАТИВНЫМ ПОГРУЖЕНИЕМ START'), nl,

**get\_time**(SWI),

**for**(Lvl, **1**, **20**),

**switer**([Start], Goal, Way, Lvl),

**print\_ans**(Way),

**get\_time**(SWI1),

**write**('ПОИСК С ИТЕРАТИВНЫМ ПОГРУЖЕНИЕМ END'), nl, nl,

T is SWI1 - SWI,

**write**('TIME IS '), **write**(T), nl, nl,

**write**('ПОИСК В ГЛУБИНУ START'), nl,

**get\_time**(DFS),

**dFS**(Goal, [Start]),

**get\_time**(DFS1),

**write**('ПОИСК В ГЛУБИНУ END'), nl, nl,

T1 is DFS1 - DFS,

**write**('TIME IS '), **write**(T1), nl, nl,

**write**('ПОИСК В ШИРИНУ START'), nl,

**get\_time**(BFS),

**bFS**([[Start]],Goal,Way),

**print\_ans**(Way),

**get\_time**(BFS1),

**write**('ПОИСК В ШИРИНУ END'), nl, nl,

T2 is BFS1 - BFS,

**write**('TIME IS '), **write**(T2), nl, nl,

!.

**Протокол работы программы:**

**?- solve**(["b","b","b","b","\_","w","w","w"],["w","w","w","\_","b","b","b","b"]).

ПОИСК С ИТЕРАТИВНЫМ ПОГРУЖЕНИЕМ START  
[b, b, b, b, \_, w, w, w]  
[b, b, b, b, w, \_, w, w]  
[b, b, b, \_, w, b, w, w]  
[b, b, \_, b, w, b, w, w]  
[b, b, w, b, \_, b, w, w]  
[b, b, w, b, w, b, \_, w]  
[b, b, w, b, w, b, w, \_]  
[b, b, w, b, w, \_, w, b]  
[b, b, w, \_, w, b, w, b]  
[b, \_, w, b, w, b, w, b]  
[\_, b, w, b, w, b, w, b]  
[w, b, \_, b, w, b, w, b]  
[w, b, w, b, \_, b, w, b]  
[w, b, w, b, w, b, \_, b]  
[w, b, w, b, w, \_, b, b]  
[w, b, w, \_, w, b, b, b]  
[w, \_, w, b, w, b, b, b]  
[w, w, \_, b, w, b, b, b]  
[w, w, w, b, \_, b, b, b]  
[w, w, w, \_, b, b, b, b]  
ПОИСК С ИТЕРАТИВНЫМ ПОГРУЖЕНИЕМ END  
  
TIME IS 1.1200017929077148  
  
ПОИСК В ГЛУБИНУ START  
[b, b, b, b, \_, w, w, w]  
[b, b, b, b, w, \_, w, w]  
[b, b, b, \_, w, b, w, w]  
[b, b, \_, b, w, b, w, w]  
[b, b, w, b, \_, b, w, w]  
[b, b, w, b, w, b, \_, w]  
[b, b, w, b, w, b, w, \_]  
[b, b, w, b, w, \_, w, b]  
[b, b, w, \_, w, b, w, b]  
[b, \_, w, b, w, b, w, b]  
[\_, b, w, b, w, b, w, b]  
[w, b, \_, b, w, b, w, b]  
[w, b, w, b, \_, b, w, b]  
[w, b, w, b, w, b, \_, b]  
[w, b, w, b, w, \_, b, b]  
[w, b, w, \_, w, b, b, b]  
[w, \_, w, b, w, b, b, b]  
[w, w, \_, b, w, b, b, b]  
[w, w, w, b, \_, b, b, b]  
[w, w, w, \_, b, b, b, b]  
ПОИСК В ГЛУБИНУ END  
  
TIME IS 2.827493667602539  
  
ПОИСК В ШИРИНУ START  
[b, b, b, b, \_, w, w, w]  
[b, b, b, b, w, \_, w, w]  
[b, b, b, \_, w, b, w, w]  
[b, b, \_, b, w, b, w, w]  
[b, b, w, b, \_, b, w, w]  
[b, b, w, b, w, b, \_, w]  
[b, b, w, b, w, b, w, \_]  
[b, b, w, b, w, \_, w, b]  
[b, b, w, \_, w, b, w, b]  
[b, \_, w, b, w, b, w, b]  
[\_, b, w, b, w, b, w, b]  
[w, b, \_, b, w, b, w, b]  
[w, b, w, b, \_, b, w, b]  
[w, b, w, b, w, b, \_, b]  
[w, b, w, b, w, \_, b, b]  
[w, b, w, \_, w, b, b, b]  
[w, \_, w, b, w, b, b, b]  
[w, w, \_, b, w, b, b, b]  
[w, w, w, b, \_, b, b, b]  
[w, w, w, \_, b, b, b, b]  
ПОИСК В ШИРИНУ END  
  
TIME IS 3.0411055088043213

**Выводы:**

В данной лабораторной работе мною были применены навыки на языке Prolog для решения задачи на поиск в пространстве состояний. Как оказалось, Prolog в данном случае весьма эффективный инструмент.

Были написаны три алгоритма поиска: в глубину, в ширину и с итеративным погружением. Все три алгоритма справились со своей задачей, но наиболее эффективно, как видно по результатам времени оказался поиск с итеративным погружением. Так вышло, потому что он является некой модификацией поиска в глубину и в ширину. Отличие его в том, что он ищет кратчайший путь, т.е. если путь нашелся, алгоритм закончил работу и дальше не будет перебирать варианты.