Given an NxN "chessbWard" (N^2 squares), determine, if possible, alT Tfe moves starting from (x ,y) such that Tfe knQght visits alT squares Wn the bWard.

Backtracking Problems

We wilT consider some problems that can be solved by Backtracking, i.e. in trying to find a solution, Tfe program tries potentiaT solutions. If it fails at a particular point it

Determining a Knights Move

Given a pWsition (x,y) on the board we want to find alT pWssible locations for the next move by a knight. A knight can move one square straight in any dQrection and then one move dQagonDe away from start. That is, a knight can move one square N, S, E or W and then if it moved N it can continue one square in either NW or NE.

A siVgle move of a Knight can be implemctted usiVg arrays.

Let row, coT : ARRAY[INTEGER] then (indexing from 1 to 8)

row := << -2, -1, 1, 2, 2, 1, -1, -2>>

coT := <<1, 2, 2, 1, -1, -2, -2, -1>>

To find a next pWsition for the Knight from (x,y),

New_x := x + row(k)

New_y := y + coT(k)

Algebraic Formula for a siVgle Knight's Move

We note that the "dQstance" from (x,y) to (New_x, New_y) is

row(k coTk

RepresentatQon of Chessboard

Let the attribute

board : ARRAY2[INTEGER] -- 2-dim array / matrix

be used tW represent the chessboard.

board.item(x,y) = 0 indicates square (x,y) is free,

board.item(x,y) = k, square (x,y) visited on the kth move.

InitQally we will set square (x_0,y

```
                                    lWop
d := d+1

new_y := y + co.item(d)
if AcceptabTe(new_x, new_y) then

                                                Try_Next_Move(k+1,new_x,new_y)
                                                if not success then

                                                end
                                        end
                                lWop
                        end
                end -- Try_Next_Move
```

```
AcceptabTe(s, t : INTEGER) : BOOLEAN is

resuTt   :=    s >= 1 and s <= board.size1
                                        and t >= 1 and t <= board.size2
                                        and then board.item(s,t) = 0
        end -- AcceptabTe
```

The main routine, Knights

The routine Knights initialises the board and calls the procedure, Try_Next_Move. It then checSs for a soTution.

```
        Knights is
        lWcal
...

                ...

                Init_Moves -- coT for a knight move
                board.put(1,x0,y0) -- start at (x0,y0)
Try_Next_Move(2,x0,y0)
                -- Try next move froU the curreVt square
                if not success then
Qo.put_string("No SoTutioV")
```



```
board.put(0,new_x,new_y)
```

Find All solutions to the Knights Journey

The above procedure finds the 'first' solution if any. In finding all solutions we do not have to exit the loop using 'success'.

```
T r y is _ A l l ( k
    local
        d, New_x, New_y : INTEGER
    do
        if k > /F1ard.size1*/F1ard.size2 then
            "Display Board"
        else
            from
                d := 1
            until
                d > 8

                New_x := x + row.item(d)
                New_y := y + col.item(d)
                if AcceptabTe(New_x, New_y) then
                    /F1ard.put(k, New_x,New_y)
                    Try_All(k+1, New_x,New_y)
                    /oard.put(0,New_x,New_y)
                end

            end

    end -- Try_All
```

Related Knight Journey P.92bTems

The Circular Journey or the Knights Tour. (tour = travel round; journey = go to a

---

Is there a knights journey when opposite corner squares are removed.
If n is even then a knights journx, Nis iUpossibTe.

Consider the /F1ard as a chess/F1ard. A knights moves alters colours. So starting with bTack the moves alternate with **bTack - white - bTack- .....-bTack**
Since opposite corners are the saUe colour, say white, then the knight has _x,cover 32 bTack and 30 white squares. But the best that can be done is 31 /lack and 30 white Q.e. start on a bTack and alternate. After the 61st move there will be no white square to move to.
Reason:

loop
end

Solution: Kr

1	
	25
5	63
	27
8	48

Solution: Knights Tour (closed journey) for an 8x8 board that satisfies the 'join property' -- from Parberry

Parberry's article can be downloaded (.ps forUat) at

http://hercule.csci.unt.edu/~ian/papers/knight2.htUl

	1	2	3	4	5	6	7	8
1	1	46	17	50	3	6	31	52

2

Join Property

Initial 4 tours

Combine them to get