

# The Eight Queens Problem

Q							
						Q	
				Q			
							Q
	Q						
			Q				
					Q		
		Q					

safe(Boardsize, Configuration) :-

    buildupto(Boardsize, RevConfig, 0, []),  
    reverse(RevConfig, Configuration).

buildupto(Boardsize,  
    Configuration,  
        Boardsize,  
            Configuration).

buildupto(Boardsize,  
    Configuration,  
        Currentcol,  
            PartialConfig) :-  
    Newcol is Currentcol+1,  
    ibt(1, Boardsize, Newrow),  
    noclash([Newrow, Newcol], PartialConfig),  
    buildupto(Boardsize,  
        Configuration,  
            Newcol,  
                [[Newrow, Newcol] | PartialConfig])

**noclash([I,J], []).**

**noclash([I,J], [[K,L] | X]) :-**

**I \== K, % J \== L, cols always \==**

**D1 is I-J, D2 is K-L, D1 \== D2,**

**S1 is I+J, S2 is K+L, S1 \== S2,**

**noclash([I,J], X).**

**ibt(L, H, I) :- number(I), !, L =< I, I =< H.**

**ibt(L, H, I) :- I is L.**

**ibt(L, H, I) :- L1 is L+1, L1 =< H, ibt(L1, H, I).**

**reverse([], []).**

**reverse([H | T], W) :-**

**reverse(T, U), append(U, [H], W).**

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

**append([H | T], U, [H | W]) :- append(T, U, W).**

## Choosing New Rows Wisely

```
safe(Bdsize, Config) :-
```

```
    clearrows,
```

```
    buildupto(Bdsize, RevConfig, 0, []),
```

```
    reverse(RevConfig, Config).
```

```
buildupto(Bdsize, Config, Bdsize, Config).
```

```
buildupto(Bdsize, Config, CCol, PartialConfig) :-
```

```
    NCol is CCol+1,
```

```
    ibt(1, Bdsize, NRow),
```

```
    not taken(NRow),
```

```
    nodiag([NRow, NCol], PartialConfig),
```

```
    asstretr(taken(NRow)),
```

```
    buildupto(Bdsize,
```

```
        Config,
```

```
        NCol,
```

```
        [[NRow, NCol] | PartialConfig])).
```

```
asstretr(taken(R)) :- not taken(R), asserta(taken(R));
```

```
    retract(taken(R)), fail.
```

```
clearrows :- asserta(taken(0)),
```

```
    setof(R, retract(taken(R)), L).
```

```
nodiag([I,J], []).
```

```
nodiag([I,J], [[K,L] | X]) :- D1 is I-J, D2 is K-L, D1\==D2,
```

```
    S1 is I+J, S2 is K+L, S1\==S2,
```

```
    nodiag([I,J], X).
```