

Spanning Tree Solution outline

```

Program    Spanning Tree
declare    T, G : array[1..N, 1..N] of boolean
              r : integer
always
  < [] i : 1 ≤ i ≤ N ::
      tree(i) = (i = r) ∨ ( ∃ j : 1 ≤ j ≤ N :: T(j,i) )
initially
  1 ≤ r ≤ N
  [] T = 0
  [] < [] p,q : (1,1) ≤ p,q ≤ (N,N) :: G(p,q) = G(q,p) >

assign
  < [] p,q : (1,1) ≤ p,q ≤ (N,N) ∧ G(p,q) ::
      T(p,q) := 1 if tree(p) ∧ ¬tree(q)
  >
end

```

Stratego Solution outline

```

Program    Stratego
declare    G : array[1..N, 1..N] of integer range 0..K
always
  < [] p,q : (1,1) ≤ p,q ≤ (N,N) ::
      count(p) = < + i : (1,1) ≤ i ≤ (N,N) ∧ G(i)=G(p) ≠ 0 :: 1 > >
      [] neighbor(p,q) = p ≠ q ∧ |p-q| ≤ (1,1)
      [] capture(p,q) = neighbor(p,q) ∧ G(p) > G(q) = 0
      [] winner(p,q) = neighbor(p,q) ∧ count(p) > count(q) > 0
  >

initially

assign
  < [] p,q : (1,1) ≤ p,q ≤ (N,N) ::
      G(q) := G(p) if capture(p,q)
      || < [] i : (1,1) ≤ i ≤ (N,N) ::
          G(i) := GP(p) if winner(p,q) ∧ G(i)=G(q) >
  >
end

```