```
procedure Move( x, d : ℕ);
local j : ℕ;
begin
    j := π^{-1}[x]; π[j] := π[j + d]; π[j + d] := x;
    π^{-1}[x] := j + d; π^{-1}[π[j]] := j;
end {of Move};

procedure Perm ( n : ℕ);
local i : ℕ;
begin
    if n > N then PrintIt
    else
        Perm( n+1 );
        for i := 1 to n − 1 do
            Move( n, dir[n] ); Perm( n + 1 );
    dir[n] := −dir[n];
end {of Perm};
```

Algorithm 5.11: Recursive ~~Pascal~~ implementation of the SJT Algorithm.

```
procedure Next;
{Assumes that π_0 = π_{n+1} = n + 1.}
begin
    m := n;
    while π[π^{-1}[m] + d[m]] > m do          {Find largest mobile integer}
        d[m] := −d[m];   m := m − 1;
    π[π^{-1}[m]] :=: π[π^{-1}[m] + d[m]]; {update π}
    π^{-1}[π[π^{-1}[m]]] :=: π^{-1}[m];       {update π^{-1}}
end {of Next};
```

Algorithm 5.12: Iterative Next implementation of the SJT algorithm (assumes $\pi_0 = \pi_{n+1} = n + 1$).

stop at $n = 1$?

```
procedure Permute ( m : integer );
var i : integer;
begin
    if m = 0 then PrintIt;
    for i := 1 to m do
        π_i :=: π_m;
        Permute( m − 1 );
        π_i :=: π_m;
end {of Permute};
```