```
select_upto  THM

⊢  ∀m:ℕ, n:ℕm. upto(m)[n] = n
| by InductionOnNat THEN RecUnfold ‘upto‘ 0 THEN SplitOnConclITE
|    THEN
|    Decide (n<m-1)
|
|\
| 1. m : ℤ
| 2. 0<m
| 3. ∀n:ℕ(m-1). upto(m-1)[n] = n
| 4. n : ℕm
| 5. ¬m = 0
| 6. n<m-1
| ⊢  (upto(m-1) @ [(m-1)])[n] = n
|   by RWO
|      Thm* ∀as,bs:T List, i:ℕ||as||. (as @ bs)[i] = as[i]
|      {select_append_front}
|      0
|      THEN
|      Try (RWO Thm* ∀n:ℕ. ||upto(n)|| ∼ n    {length_upto} 0)
|      THEN
|      Try BackThruSomeHyp
|
.\
. 1. m : ℤ
. 2. 0<m
. 3. ∀n:ℕ(m-1). upto(m-1)[n] = n
. 4. n : ℕm
. 5. ¬m = 0
. 6. ¬n<m-1
. ⊢  (upto(m-1) @ [(m-1)])[n] = n
. | by RWO
. |    Thm* ∀as,bs:T List, i:{||as||..(||as||+||bs||)⁻}.
. |         (as @ bs)[i] = bs[(i-||as||)]
. |    {select_append_back}
. |    0
. |    THEN
. |    Try (RWO Thm* ∀n:ℕ. ||upto(n)|| ∼ n    {length_upto} 0)
. |
. 1. m : ℤ
. 2. 0<m
. 3. ∀n:ℕ(m-1). upto(m-1)[n] = n
. 4. n : ℕm
. 5. ¬m = 0
. 6. ¬n<m-1
. ⊢  [(m-1)][(n-(m-1))] = n  by Subst’ (n-(m-1) = 0) 0 THEN Reduce 0
END
```