1 Selection Sort

Selection sort is a sorting algorithm that sorts an array a by employing the following strategy: First, it finds the index m of the maximum element in the initial segment of size b of the array a using a function *findmax*. Then it swaps the elements at the index of the maximum with the element at the end of the subarray. It repeats the process until the subarray is of length 1 or less.

1. (60 points) : Implement the selection sort algorithm using mapcode. You need to define three spaces I, A and X and three maps $\rho : I \to X$, $F : X \to X$ and $\pi : X \to A$. Assume the primitives

    1. Basic arithmetic

    2. $swap(a, i, j)$ which swaps elements at i and j of the array a. You do NOT need to show its implementation.

    3. *findmax*: You do NOT need to show its implementation.

2. (15 points) : Trace the mapcode solution for the instance $a = [7, 3, 9, 8]$.

3. (15 points) : In the definition of F, you would have used *findmax* and *swap* as primitives. Draw a separate diagram that traces the details of the computation of the first iteration of F in the trace of selection sort.

   Hint:

    1. In class, we covered how to trace arbitrary computations that have subcomputations that need to be combined using pairing. You will need to use pairing. Here is an example of a computation that employs pairing: