Write a function `findExtreme` that takes a function (with type int $\times$ int $\rightarrow$ bool) and a list of integers and uses the function to select the extreme element (least, greatest, etc) of the list. Specifically, the function that `findExtreme` takes as a parameter defines a way to order int, that is, it compares two ints (say *a* and *b*) and returns `true` if *a* comes before *b* and `false` otherwise (mathematically, this function is a *total order*). Thus `findExtreme` is a generalization of `findGreatest`. For example, `findExtreme(fn (a, b) => a > b, [6, 4, 18, 9, 2])` would return 18. (This problem is *not* naturally solved using `map` or `filter`.)