```
# should always be specified for custom array/tensor types
function similarstructure_from_indices end

# generic definition, net very efficient, provide more efficient version if possible
memsize(a::Any) = Base.summarysize(a)

# generic definitions, should be overwritten if your array/tensor type does not
  support
# Base.similar(object, eltype, structure)
function similar_from_structure(A, T, structure)
    if isbits(T)
        similar(A, T, structure)
    else
        fill!(similar(A, T, structure), zero(T)) # this fixes BigFloat issues
    end
end

function similar_from_indices(T::Type, p1::IndexTuple, p2::IndexTuple, A,
CA::Symbol)
    structure = similarstructure_from_indices(T, p1, p2, A, CA)
    similar_from_structure(A, T, structure)
end
function similar_from_indices(T::Type, poA::IndexTuple, poB::IndexTuple,
                                p1::IndexTuple, p2::IndexTuple,
                                A, B, CA::Symbol, CB::Symbol)
    structure = similarstructure_from_indices(T, poA, poB, p1, p2, A, B, CA, CB)
    similar_from_structure(A, T, structure)
end

# should work generically but can be overwritten
Base.@pure function similartype_from_indices(T::Type, p1, p2, A, CA)
    Core.Compiler.return_type(similar_from_indices,
                                Tuple{Type{T}, typeof(p1), typeof(p2), typeof(A),
Symbol})
end
Base.@pure function similartype_from_indices(T::Type, poA, poB, p1, p2, A, B, CA,
CB)
    Core.Compiler.return_type(similar_from_indices,
                                Tuple{Type{T}, typeof(poA), typeof(poB),
                                        typeof(p1), typeof(p2), typeof(A),
typeof(B),
                                        Symbol, Symbol})
end

# generic, should probably not be overwritten
function cached_similar_from_indices(sym::Symbol, T::Type,
                                        p1::IndexTuple, p2::IndexTuple,
                                        A, CA::Symbol)
    if use_cache()
        structure = similarstructure_from_indices(T, p1, p2, A, CA)
        typ = similartype_from_indices(T, p1, p2, A, CA)
        key = (sym, taskid(), typ, structure)
        C::typ = get!(cache, key) do
            similar_from_indices(T, p1, p2, A, CA)
        end
        return C
    else
        return similar_from_indices(T, p1, p2, A, CA)
    end
end
function cached_similar_from_indices(sym::Symbol, T::Type,
                                        poA::IndexTuple, poB::IndexTuple,
                                        p1::IndexTuple, p2::IndexTuple,
                                        A, B, CA::Symbol, CB::Symbol)

    if use_cache()
        structure = similarstructure_from_indices(T, poA, poB, p1, p2, A, B, CA, CB)
        typ = similartype_from_indices(T, poA, poB, p1, p2, A, B, CA, CB)
        key = (sym, taskid(), typ, structure)
        C::typ = get!(cache, key) do
            similar_from_indices(T, poA, poB, p1, p2, A, B, CA, CB)
        end
        return C
    else
        return similar_from_indices(T, poA, poB, p1, p2, A, B, CA, CB)
    end
end
```