Lam Nguyen

The C and Haskell implementations pf the same algorithms differ because :

* In C you give instructions to the computer. You tell it: do this, then do that, then do that. The computer simply follows your instruction one by one until it runs out of instructions. While executing them, it can change state. In Haskell, you don't tell the computer what to do as such but rather you tell it what stuff is. You tell it how things are defined, and the compiler figures out what to call when based on your definitions. You also can't set a variable to something and then set it to something else later. Therefore, Haskell does not support loop. However, we can use loop in C.
* In Haskell, a function has no side-effects. The only thing a function can do is calculate something and return it as a result.
* Haskell programs are usually shorter than other imperative programming languages (C, C++, Java, Python …)
* In C, you have to define the type of each variables. However, Haskell has a very sophisticated type inference system. It is able to figure out that the parameter has to be an integer, string, …
* In C, you manage the memory using malloc and free. Every malloc must have a corresponding free call. In Haskell, memory management is automatic and essentially works behind the scene, you never have to call malloc or free, memory is allocated and deallocated.