bit-algorithms Documentation

Vincent Reverdy Bryce Kille Collin Gress

May 8, 2019

Contents

1 Introduction

Here we will document algorithms and their progress. Sections will be divided according to how they are in the reference. Each algorithm subsection will be introduced with a description of the algorithms behavior and signature, followed by the current implementation pseudocode in `bit`. Any other possible implementations may follow. At the end, a there can be a discussion of potential optimization, drawbacks, or bugs for each implementation.

2 Example operations

2.1 `max`

```
template< class T >
constexpr const T& max( const T& a, const T& b );  (1)

template< class T, class Compare >
constexpr const T& max( const T& a, const T& b, Compare comp );  (2)

template< class T >
constexpr T max( std::initializer_list<T> ilist );  (3)

template< class T, class Compare >
constexpr T max( std::initializer_list<T> ilist, Compare comp );  (4)
```

Returns the greater of the given values.

1-2) Returns the greater of `a` and `b`.

3-4) Returns the greatest of the values in intializer list `ilist`.

The (1,3) versions use operator `<` to compare the values, the (2,4) versions use the given comparison function `comp`.

Parameters

Parameters go here. TODO find a better way to format these

a, b	-	the values to compare.
ilist	-	initializer list with the values to compare.
comp	-	comparison function object

Type requirements

Return value

Complexity

Implementation

Possible Implementations

Notes

3 Non-modifying sequence operations

4 Modifying sequence operations