```rust
#![feature(test)]
#[macro_use]
extern crate abomonation;
extern crate test;

use test::Bencher;
use abomonation::{Abomonation, encode, decode};

#[bench]
fn bench_populate(b: &mut Bencher) {
    b.iter(|| {
        Log::new()
    });
}

#[bench]
fn bench_serialize(b: &mut Bencher) {
    let log = Log::new();
    let mut bytes = vec![];
    unsafe { encode(&log, &mut bytes); }
    b.bytes = bytes.len() as u64;
    b.iter(|| {
        bytes.clear();
        unsafe { encode(&log, &mut bytes); }
        test::black_box(&bytes);
    });
}

#[bench]
fn bench_deserialize(b: &mut Bencher) {
    let log = Log::new();
    let mut bytes = vec![];
    unsafe { encode(&log, &mut bytes); }
    b.bytes = bytes.len() as u64;
    b.iter(|| {
        test::black_box(unsafe { decode::<Log>(&mut bytes) });
    });
}
```

```
bench_populate:           267 ns/iter (+/- 124)
bench_serialize:           54 ns/iter (+/- 4)  = 10148 MB/s
bench_deserialize:          8 ns/iter (+/- 1)  = 68500 MB/s
bench_deserialize_test:   112 ns/iter (+/- 25) =  4892 MB/s
```

```
22      b.iter(|| {
23          bytes.clear();
24          unsafe { encode(&log, &mut bytes); }
25          test::black_box(&bytes);
26      });
27  }
28
29  #[bench]
30  fn bench_deserialize(b: &mut Bencher) {
31      let log = Log::new();
32      let mut bytes = vec![];
33      unsafe { encode(&log, &mut bytes); }
34      b.bytes = bytes.len() as u64;
35      b.iter(|| {
36          test::black_box(unsafe { decode::<Log>(&mut bytes) });
37      });
38  }
```

```
bench_populate:          267 ns/iter (+/- 124)
bench_serialize:          54 ns/iter (+/- 4)  = 10148 MB/s
bench_deserialize:         8 ns/iter (+/- 1)  = 68500 MB/s
bench_deserialize_test:  112 ns/iter (+/- 25) =  4892 MB/s
```

```
22          b.iter(|| {
23              bytes.clear();
24              unsafe { encode(&log, &mut bytes); }
25              test::black_box(&bytes);
26          });
27      }
28
29      #[bench]
30      fn bench_deserialize(b: &mut Bencher) {
31          let log = Log::new();
32          let mut bytes = vec![];
33          unsafe { encode(&log, &mut bytes); }
34          b.bytes = bytes.len() as u64;
35          b.iter(|| {
36              test::black_box(unsafe { decode::<Log>(&mut bytes) });
37          });
38      }
```

```
bench_populate:         267 ns/iter (+/- 124)
bench_serialize:         54 ns/iter (+/- 4)  = 10148 MB/s
bench_deserialize:        8 ns/iter (+/- 1)  = 68500 MB/s
```