


```
1  #![feature(test)]
2  #[macro_use]
3  extern crate abomonation;
4  extern crate test;
5
6  use test::Bencher;
7  use abomonation::{Abomonation, encode, decode};
8
9  #[bench]
10 fn bench_populate(b: &mut Bencher) {
11     b.iter(|| {
12         Log::new()
13     });
14 }
15
16 #[bench]
17 fn bench_serialize(b: &mut Bencher) {
18     let log = Log::new();
19     let mut bytes = vec![];
20     unsafe { encode(&log, &mut bytes); }
21     b.bytes = bytes.len() as u64;
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


```
1  #![feature(test)]
2  #[macro_use]
3  extern crate abomonation;
4  extern crate test;
5
6  use test::Bencher;
7  use abomonation::{Abomonation, encode, decode};
8
9  #[bench]
10 fn bench_populate(b: &mut Bencher) {
11     b.iter(|| {
12         Log::new()
13     });
14 }
15
16 #[bench]
17 fn bench_serialize(b: &mut Bencher) {
18     let log = Log::new();
19     let mut bytes = vec![];
20     unsafe { encode(&log, &mut bytes); }
21     b.bytes = bytes.len() as u64;
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 }
```

benchmark_initialize: 8ns/iter (+/- 1) = 68500 MB/s

bench_serialize: 54ns/iter (+/- 4) = 10148 MB/s

benchmark serialized test: 112ns / iter (+/- 25) = 4892 MB/s

bench_populate:267ns/iter (+/- 124)