**Benchtest的简单使用**

一个简单的benchtest用例

// 以BenchmarkXXX类似命名，并传入b \*testing.B 参数

func BenchmarkLoopSum(b \*testing.B) {

for i := 0; i < b.N; i++ {

total := 0

for j := 0; j <= maxLoop; j++ {

total += j

}

}

}

查看benchtest的参数: go help testflag  
-bench grep  
通过正则表达式过滤出需要进行benchtest的用例  
-count n  
跑n次benchmark，n默认为1  
-benchmem  
打印内存分配的信息  
-benchtime=5s  
自定义测试时间，默认为1s

测试命令：$ go test --bench=LoopSum my\_test.go -benchmem  
运行结果:

goos: windows

goarch: amd64

BenchmarkLoopSum-12 5000 316952 ns/op 0 B/op 0 allocs/op

PASS

ok command-line-arguments 1.939s

5000表示测试次数，即test.B提供的N, ns/op表示每一个操作耗费多少时间(纳秒)。B/op表示每次调用需要分配16个字节。allocs/op表示每次调用有多少次分配  
基准测试框架对一个测试用例的默认测试时间是 1 秒。开始测试时，当以 Benchmark 开头的基准测试用例函数返回时还不到 1 秒，那么 testing.B 中的 N 值将按 1、2、5、10、20、50……递增，同时以递增后的值重新调用基准测试用例函数。

$ go test --bench=. my\_test.go -benchmem

goos: windows

goarch: amd64

BenchmarkRange-12 100000 20505 ns/op 8 B/op 0 allocs/op

BenchmarkFor-12 1000000 2054 ns/op 0 B/op 0 allocs/op

BenchmarkLoopSum-12 5000 315755 ns/op 0 B/op 0 allocs/op

BenchmarkLoopRecursion-12 300 4664190 ns/op 0 B/op 0 allocs/op

PASS

ok command-line-arguments 8.792s