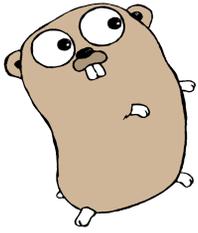


# New Language, Go, Promising for Scientific Programming



Sonia Keys

Smithsonian Astrophysical Observatory  
skeys@cfa.harvard.edu



Potential use of Go, a free and open source computer language sponsored by Google, is explored by porting a small program and comparing to other languages. Ported was Digest2, a program in use at the Minor Planet Center to screen submitted but unidentified asteroid astrometry tracklets for potential detections of Near Earth Objects. Go seems a promising new choice for scientific programming. It was found to have execution speed comparable to Fortran and C, source code simplicity comparable to Python, and a concurrency model that is unique and intuitive.

```
package main
import (
    "container/vector"
    "flag"
    "fmt"
    "gob"
    "io/ioutil"
    "os"
    "rand"
    "runtime"
    "strings"
)
const (
    usage = "Usage: digest2 <obsfile>"
    gfn = "digest2.pop"
)
func main() {
    flag.Parse()
    if flag.NArg() != 1 {
        fmt.Println(usage)
        return
    }
    // must read obscode.dat before observation
    if err := read_obs(); err != nil {
        fmt.Println("Error reading", ofn)
        return
    }
    // while we're checking prerequisites, read
    if err := read_pop(); err != nil {
        fmt.Println("Error reading", gfn)
        return
    }
    // prerequisites met. read observations.
    tracklets, err := read_obs(flag.Arg(0))
    if err != nil {
        fmt.Println("Error reading", flag.Arg(0))
        return
    }
    fmt.Println("Desig. Raw No-ID V-mag")
    distribute(tracklets)
}
var (
    ocdMap map[string]*ocd
)
all_ss []][5][18][25]float64
unk_ss []][5][18][25]float64
all_mpcint []][5][18][25]float64
unk_mpcint []][5][18][25]float64
// read_pop reads population model (created by muk)
func read_pop() os.Error {
    f, err := os.Open(gfn, os.O_RDONLY, 0)
    if err != nil {
        return err
    }
    defer f.Close()
    dec := gob.NewDecoder(f)
    if err = dec.Decode(&ocdMap); err != nil {
        return err
    }
    if err = dec.Decode(&all_ss); err != nil {
        return err
    }
    if err = dec.Decode(&unk_ss); err != nil {
        return err
    }
    if err = dec.Decode(&all_mpcint); err != nil {
        return err
    }
    if err = dec.Decode(&unk_mpcint); err != nil {
        return err
    }
    return nil
}
type observation struct {
    mjd float64
    ra float64
    dec float64
    magV float64
    site *ocd
    earth_observer [3]float64
}
type tracklet struct {
    desig string
    obs observation
    magV float64
    rch string
    rnd float64
    randMpcintScore, noID bool
    tags string
    tag int
    sumAllSS, sumUnkSS, sumAllMpcint, sumUnkMpcint float64
}
// distance independent working variables. computed over the course
// of processing the tracklet.
dt, invdt, invdtsq float64
sun_observer [2][3]float64 // vectors at times t0 and t1
observer_object_unit [2][3]float64 // vectors at times t0 and t1
observer_object0 [3]float64
observer_object0_mag float64
sun_object0 [3]float64
sun_object0_mag float64
sun_object0_magsq float64
observer1_object0 [3]float64
observer1_object0_mag float64
observer1_object0_magsq float64
tz, hmag float64
hmagBin int
dTags int
dTag [24][5][18][25]bool
// distribute tracklet processing across multiple cores
func distribute(tracklets []*tracklet) {
    // the number chosen is not terribly critical. it's nice if it is big
    // enough to keep all cores busy. having it bigger just increases memory
    // requirements and makes for a little more thread swapping. GOMAXPROCS
    // is the best guess offered by the current go scheduler.
    nWorkers := runtime.GOMAXPROCS(0)
    if len(tracklets) < nWorkers {
        nWorkers = len(tracklets)
    }
    // tkch is used to dispatch tracklets to workers. all workers get
    // tracklets from this single channel.
    tkch := make(chan *tracklet)
    for i := 0; i < nWorkers; i++ {
        go solve(tkch)
    }
    // prch is used to keep processed results in submission order.
    // it is a buffered channel so that a fast process can drop off the
    // result without waiting for processes ahead of it. the size of
    // the buffer must be at least nWorkers, but otherwise isn't critical.
    // Having it somewhat larger allows more results to back up behind
    // a slow process. We expect processing time to not vary too much
    // anyway.
    prch := make(chan chan string, nWorkers*2)
    // dispatcher communicates synchronously with workers and so needs
    // a separate goroutine
    go dispatch(tracklets, tkch, prch)
    // everything is on it's way. just wait for results and print them
    // as they are available. prch is our buffer of result channels in
    // the correct order.
    for rch := range prch { // wait here for the next channel in order
        fmt.Println(<-rch) // wait here for processing result
    }
}
```

### Go<sup>11</sup> Synopsis

**Simplicity**

- Clean library mechanism with no header files.
- Type inferencing.
- Complexity of modern object oriented languages consciously avoided.

**Safety**

- Strong static typing.
- Garbage collection.
- No pointer arithmetic.

**Concurrency**

- Based on CSP<sup>12</sup>, but innovative in a new direction.
- "Goroutines" have runtime multiplexed threading.
- "Channels" are analogous to Unix pipes, but in memory and more flexible. Synchronous and asynchronous communication, buffered and unbuffered, receipt checked or unchecked, multiple reader, multiple writer. Channels are typed first class objects.

```
const (
    twoPi = 2 * math.Pi
    MIN_DISTANCE = .05
    MAX_DISTANCE = 7
    K = .01720209895
    INV_K = 1 / K
    U = K * K
)
// local sidereal time
func lst(j0, longitude float64) float64 {
    t := (j0 - 15019.5) / 36525
    th := (6.6460656 + (2400.051262 + 0.00002581*t)*t) / 24
    ut := math.Fmod(1, j0 - 5)
    return math.Fmod(th+ut+longitude, twoPi)
}
// compute a1 - a2
func sub3(a1 [3]float64, a2 [3]float64) {
    a1[0] -= a2[0]
    a1[1] -= a2[1]
    a1[2] -= a2[2]
}
/* eRotate
rotate to ecliptic coordinates.
Args:
    c: vector in equatorial coordinates.
    soe, coe: sine, cosine of ecliptic.
Side effect:
    c rotated in place.
*/
func eRotate(c [3]float64, soe, coe float64) {
    e1 := c[2]*soe + c[1]*coe
    c[2] = c[2]*coe - c[1]*soe
    c[1] = e1
}
// vector dot product
func dot3(a1, a2 [3]float64) float64 {
    return a1[0]*a2[0] + a1[1]*a2[1] + a1[2]*a2[2]
}
/* cross3
vector cross product
Args:
    a, b: three element vectors of double
Returns:
    a x b
*/
func cross3(a, b [3]float64) (result [3]float64) {
    result[0] = a[1]*b[2] - a[2]*b[1]
    result[1] = a[2]*b[0] - a[0]*b[2]
    result[2] = a[0]*b[1] - a[1]*b[0]
    return result
}
/* tagAngle (tracklet method)
process a single distance-angle combination.
Args:
    tk: tracklet with distance setup already done.
    an: angle for orbit solution
Notes:
    solves orbit for passed angle, converts to bin indicies, sets bin tag
    and updates tag count.
*/
func tagAngle(tk *tracklet, d2 := tk.observer1_0) {
    // velocity scaled b
    var v [3]float64
    hv := dot3(tk.sun, hv)
    hsq := dot3(hv, hv)
    hm := math.Sqrt(hsq)
    // solve for semi-major axis
    // (and the inverse)
    vsq := dot3(v, v)
    temp := 2 - tk.sun_0
    // far model, require
    // for stability, r
    if tk.sun_object0_mag
        return false
    }
    orbit_a := tk.sun_ob
    inva := temp / tk.su
    // solve for eccentricity
    // (stability test o
    orbit_e := math.Sqrt
    // stability test:
    if orbit_e > .99 {
        return false
    }
    // solve for inclina
    // reliable check fo
    // loss of precision
    lzero := hv[2] == hm
    // combination of st
    // ensure that hm is
    var orbit_i float64
    if lzero {
        orbit_i = math.Acos(hv[2]/hm) * 180 / math.Pi
    }
}
```

### Features Enabling Scientific Programming

**Easy**

- Imperative, procedural programming style is the style familiar to most scientists. It is the style of Fortran and C.
- A clean, new language design without historical baggage.

**Fast**

- Native code compilers generate fast object code. Performance is comparable to other compiled languages like Fortran and C.
- Modern concurrency features designed for today's multicore CPUs.
- IEEE floating point, native complex types.

**Accessible**

- Free and open source.
- Runs on Linux, Darwin, Windows operating systems. x86, x86-64, and ARM architectures.
- SWIG support. This allows Go to link to existing well tested and highly optimized C and C++ libraries like BLAS/LAPACK, GNU Scientific Library, FFTW.
- Two compiler suites, one a GCC front end and the other stand-alone.

```
func read_obs(ofn string) ([]*tracklet, os.Error) {
    // read observations all at once
    b, err := ioutil.ReadFile(ofn)
    if err != nil {
        return nil, err
    }
    // split by line
    lines := strings.Split(b, "\n", -1)
    Error(fmt.Sprintf("No tracklets in %s", ofn))
    tracklets := []*tracklet{}
    Error(fmt.Sprintf("No tracklets in %s", ofn))
    // split observation stream into tracklets.
    // the stream must have observations grouped
    // chronologically within each object. Lines not
    // let are unceremoniously dropped.
    for i := 0; i < len(lines); i++ {
        line := strings.TrimSpace(lines[i])
        if line == "" {
            continue
        }
        r := Vector{}
        g := gob.NewDecoder(r)
        if err := g.Decode(&tk); err != nil {
            continue
        }
        tk.design = desig
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID = parseMpcint(line)
        tk.tag = tag
        tk.sumAllSS, tk.sumUnkSS, tk.sumAllMpcint, tk.sumUnkMpcint = parseMpcint(line)
        tk.tracklet = tk
        tk.obs = obs
        tk.magV = magV
        tk.rch = rch
        tk.rnd = rnd
        tk.randMpcintScore, noID
```