

## 1.4 Quelques réflexions sur la complexité à partir d'exemples

### Parcours

```
1. for i:=1 to length(t)
2.   writeln(t[i]) ;
```

```
1. for i:=1 to n
2.   readln(a) ;
3.   b := a*9/5 + 32 ;
4.   writeln(b) ;
```

Notions : – taille de l'input  
– *random-access machine*  
– temps d'exécution – fonction linéaire *vs.* quadratique

### Recherche (linéaire)

```
1. for i:=1 to length(t)
2.   if t[i]=key then writeln(i) ;
```

```
1. i:= 1
2. while(t[i] <> key) do
3.   i := i + 1 ;
4. if t[i]=key then writeln(i) ;
```

Notions : – cas le pire et cas moyen  
– taux (ordre) de croissance

### Tri

champion:

```
1. best := t[1] ;
2. for i:= 2 to length(t)
3.   if best < t[i] then best := t[i]
```

champion(d,f) :

```
1. best := t[d] ;
2. for i:=d+1 to f
3.   if best < t[i] then best := t[i]
4. return best
```

tri :

```
1. for i:=1 to length(t)
2. t[i] := champion(i,length(t)) ;
```