

1.4 Quelques réflexions sur la complexité à partir d'exemples

Parcours

```
1. for i in range(len(t)):  
2.     print(t[i])
```

```
1. for i in range(n):  
2.     a = int(input())  
3.     b = a*9/5 + 32  
4.     print(b)
```

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

Recherche (linéaire)

```
1. for i in range(len(t)):  
2.     if t[i] == key:  
3.         print(i)
```

```
1. i = 1  
2. while (t[i] != key) :  
3.     i = i + 1  
4. if t[i]==key:  
5.     print(i)
```

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

Tri

```
# champion:  
1. best = t[1] ;  
2. for i in range(1,len(t)):  
3.     if best < t[i]:  
4.         best = t[i]
```

```
# champion sur un sous-range  
1. def champion(d,f):  
2.     best = t[d]  
3.     for i in range(d,f):  
4.         if best < t[i]:  
5.             best = t[i]  
6. return best
```

```
# Un algorithme de tri :  
1. for i in range(len(t)):  
2.     t[i] = champion(i,len(t))
```