

## *Travaux dirigés n° 1*

### Analyse des algorithmes itératifs

#### Exercice 1.1

Soit  $N$  un entier naturel, on pose  $n = \log_2(N)$  et  $m = \log_3(N)$ . Pour l’ordre de grandeur de chaque fonction, dire si cela est **Vrai** ou **Faux** :

1.  $n \in O(m)$ ;
2.  $2^n \in O(2^m)$ ;
3.  $n^2 \in O(m^2)$ ;
4.  $3^N \in O(2^N)$ .

#### Exercice 1.2

Quelle est, en fonction de  $N$ , la complexité temporelle de l’algorithme suivant ? On suppose que chaque opération se fait en temps constant  $O(1)$ .

```
TOTO (entier N) {  
1   count ← 0  
2   pour (i ← 1; i*i ≤ N; i++) {  
3       count++  
4   }  
4   retourner count  
}
```

#### Exercice 1.3

Quelle est, en fonction de  $N$ , la complexité temporelle de l’algorithme suivant ? On suppose que chaque opération se fait en temps constant  $O(1)$ .

```
FOO (entier N) {  
1   count ← 0  
2   pour (i ← N div 2; i ≤ N; i++) {  
3       pour (j ← 1; j ≤ N; j ← 2*j) {  
4           pour (k ← 1; k ≤ N; k ← 2*k) {  
5               count++  
6           }  
7       }  
8   }  
6   retourner count  
}
```

### Exercice 1.4 — Tri par insertion

L'algorithme du tri par insertion trie un tableau de  $n$  nombres. Trouvez sa complexité en fonction de  $n$ .

```
insertion_sort (entier T[], entier n) {  
1  pour (i ← 2; i ≤ n; i++) {  
2    a ← T[i]  
3    j ← i-1  
4    tant que (j > 0 et a < T[j]) {  
5      T[j + 1] ← T[j]  
6      j ← j - 1  
7    }  
8    T[j + 1] ← a  
9  }  
}
```

### Exercice 1.5 — Tri par sélection

L'algorithme du tri par sélection trie un tableau de  $n$  nombres. Trouvez sa complexité en fonction de  $n$ .

```
selection_sort (entier T[], entier n) {  
1  pour (i ← 1; i < n; i++) {  
2    minj ← i  
3    mina ← T[i]  
4    pour (j ← i+1; j ≤ n; j++){  
5      si (T[j] < mina) {  
6        minj ← j  
7        mina ← T[j]  
8      }  
9    T[minj] ← T[i]  
10   T[i] ← mina  
11 }  
}
```

### Exercice 1.6 — Questions de taille

Soit  $x$  et  $y$  deux entiers de  $n$  bits (la **taille** de  $x$  et  $y$  est donc  $n$ ). Donnez pour chacune des questions suivantes une réponse en fonction de  $n$ . Vous pouvez utiliser la notation  $O$ .

1. Quelle est la plus grande valeur de  $x$  possible ?
2. Quelle est la taille en nombre de bits de  $x^2$  ?
3. Quelle est la taille en nombre de bits de  $2^x$  ?
4. Quelle est la taille en nombre de bits de  $x \times y$  ?
5. Quelle est la taille en nombre de bits de  $x^y$  ?
6. Quelle est la taille en nombre de bits de  $x^x \bmod y$  ?
7. Quelle est la taille en nombre de bits de  $x!$  ?