

Maths pour l'Info 1

– L1 –
2013-2014

TD n° 6 : complexité de programmes

► **Exercice 1** ◀ Que fait la fonction suivante ? Quelle est sa complexité en fonction de n ?

```
void f(int n) {
    int i, j, k;

    for(i=1; i<=n; i++)
        for(j=1; j<=i; j++)
            for(k=1; k<=j; k++)
                if (i*i == j*j + k*k)
                    printf("%d %d %d \n", i, j, k);
}
```

► **Exercice 2** ◀ On considère le programme suivant :

```
int prog(int n) {
    int i;
    for(i=1; i<=n; i++)
        algo1(i);
    algo2(n);
}
```

(a) Quelle est la complexité de `prog` si `algo1` est en $\mathcal{O}(n)$ et `algo2` en $\mathcal{O}(n^3)$?

(b) Quelle est la complexité de `prog` si `algo1` est en $\mathcal{O}(n^2)$ et `algo2` en $\mathcal{O}(n^2)$?

(c) Quelle est la complexité de `prog` si `algo1` est en $\mathcal{O}(\log n)$ et `algo2` en $\mathcal{O}(n^2)$?

(d) Quelle est la complexité de `prog` si `algo1` est en $\mathcal{O}(\log n)$ et `algo2` en $\mathcal{O}(n \log n)$?

► **Exercice 3** ◀ On considère la fonction suivante pour les entiers $n \geq 2$:

```
int f(int n) {
    int i;
    for(i=2; i*i<=n; i++)
        if (n % i == 0)
            return 0;
    return 1;
}
```

(a) Que fait la fonction ?

(b) Quelle est sa complexité, sous la forme $\mathcal{O}(\cdot)$?

On considère maintenant le programme suivant :

```
int main() {
    int N, i;
    N = 1000;
    for(i=2; i<=N; i++)
        if (f(i))
            printf("%d ", i);
    return 1;
}
```

(c) Que fait ce programme ?

(d) Quelle est sa complexité, sous la forme $\mathcal{O}(\cdot)$?

► **Exercice 4** ◀ Le crible d'Eratosthène permet de calculer tous les nombres premiers plus petits que n . Le programme est le suivant :

```
void Eratosthene(int n) {
    int i, j, T[MAX];
    for(i=2; i<=n; i++)
        T[i] = 1; //a priori i premier
    for(i=2; i<=n; i++)
        if (T[i] == 1)
            for(j=2*i; j<n; j=j+i)
                T[j] = 0;
    // afficher T
}
```

Quelle est sa complexité, sous la forme $\mathcal{O}(\cdot)$?