IFT2015 - Structures de données
Démonstration - 2
22 Janvier 2014

1 Vrai ou Faux?

1. $O(n^2) = O(2^{2\log_3 n})$
2. $2^{n+1} \in O(2^n)$
3. $\log_{10} n^2 \in O(\lg n)$
4. $O(2^n) = O(3^n)$
5. $2^{2n} \in O(2^n)$
6. $f(n) \in O(n) \Rightarrow f(n) \in O(\frac{n}{\lg \lg n})$

2 À prouver!

1. $\lg(n!) \in O(n \lg n)$
2. $(n+a)^b \in O(n^b); a \in \mathbb{R}; b \in \mathbb{R}_+^*$
3. $\max(f(n), g(n)) \in \Theta(f(n) + g(n))$
4. $\log_a n + \frac{1}{\lg n} \in \Theta(\lg n)$
5. $\sum_{i=1}^{n} i^k \in \Theta(n^{k+1})$
6. Si $d(n)$ est dans $O(f(n))$, alors $a.d(n)$ est dans $O(f(n))$, pour chaque constante $a > 0$.
7. Si $d(n)$ est dans $O(f(n))$ et $f(n)$ est dans $O(g(n))$, alors $d(n)$ est dans $O(g(n))$.

3 Notation asymptotique

Donnez le temps de calcul en notation asymptotique O en fonction de n des fonctions python suivantes.

1.
```
"""Return the sum of the elements in sequence S."""
def example1(S):
        n = len(S)
        total = 0
        for j in range(n):
                # loop from 0 to n-1
                total += S[j]
        return total
```

2.
```
"""Return the sum of the elements with even index in
sequence S."""
def example2(S):

        n = len(S)
        total = 0
        for j in range(0, n, 2):
                # note the increment of 2
                total += S[j]
        return total
```

3.
```
"""Return the sum of the prefix sums of sequence S."""
def example3(S):
        n = len(S)
        total = 0
        for j in range(n): # loop from 0 to n-1
                for k in range(1+j): # loop from 0 to j
                        total += S[k]
        return total
```

4.
```
"""Return the sum of the prefix sums of sequence S."""
def example4(S):
        n = len(S)
        prefix = 0
        total = 0
        for j in range(n):
                prefix += S[j]
                total += prefix
        return total
```

5.
```
"""Return the number of elements in B equal to the sum
of prefix sums in A."""
def example5(A, B):
        # assume that A and B have equal length
        n = len(A)
        count = 0
        for i in range(n):
                # loop from 0 to n-1
                total = 0
                for j in range(n):
                        # loop from 0 to n-1
                        for k in range(1+j):
                                # loop from 0 to j
                                total += A[k]
```

```
                if B[i] == total:
                    count += 1
        return count
```