Partiel

Exercice 1 Donnez le code d'une prodédure Java
`public static double TriPasc(int n){...}`
qui déclare, alloue la mémoire, et remplie un tableau triangulaire `mat` de $\sum_{i=0}^{i=n}(i+1)$ entiers correspondant à toutes les valeurs du coefficient binomial $\binom{n}{p}$.
(Un bonus de point est prévu si vous optimisez votre code pour éviter l'usage intempestif de boucles `for`).

Exercice 2 Soit la fonction

```
public static double xx(int n, int p){
    if(p==n || p==0)
      return 1.0;
    else
      return xx(n-1,p-1)+xx(n-1,p);
  }
```

1. Que fait la fonction `xx(n,p)` ?
2. Démontrez le.
3. Soit

```
public static double yy(int n){
    for(int i=0; i<= n ; i++)
                xx(n,i)
  }
```

Montrez que la complexité de `yy(n)` est $\Theta(2^n)$.

Exercice 3 Soit un algorithme dont le temps d'exécution $T(n)$ satisfait

$$T(n) = \begin{cases} O(1) & \text{si } n \leq 1 \\ aT(n/b) + f(n) & \text{si } n \geq 2 \end{cases} \quad (a,b \text{ sont des entiers non-nuls})$$

1. Montrer par récurrence sur p que pour $n = b^p$, on a
$$T(n) = a^p + \sum_{i=0}^{i=p-1} a^i f(n/b^i)$$
2. Montrer que l'on peut remplacer ci-dessus a^p par $n^{\log_b a}$.
3. Avec $b = 2$ et $f(n) = \Theta(n^2)$, pour quelle(s) valeur(s) de a a-t-on
   (a) $T(n) = \Theta(n^3)$?
   (b) $T(n) = \Theta(n^2)$?
   (c) $T(n) = \Theta(n)$?

Exercice 4 Soit

```
public static void D(boolean[] Tab){
  int j=0;
  while(Tab[j]==true){       //j<Tab.length
    Tab[j]=false;
    j++;
  }
  Tab[j]=true;
}
public static void C(int n){
  boolean[] Tab;
  Tab = new boolean[n];   //Tab[j] == false
  for(int i=0;i<n;++i)
    D(Tab);
}
```

1. Montrer que la complexité de `C(n)` est $\Omega(n)$ et $O(n^2)$.
2. Montrer qu'elle est en fait $\Theta(n)$ en utilisant (sans le démontrer) le fait que, soit k le plus petit entier tel que 2^k n'apparait pas dans la décomposition de i en base 2, si $b_i(j) = \lfloor \frac{i}{2^j} \rfloor$ pour $i = 0$ $(\forall j)$ et pour $j = 0$ $(\forall i)$, et si $b_i(j)$ satisfait

$$b_{i+1}(j) = \begin{cases} 1 + b_i(j) & \text{si } j \leq k \text{ et} \\ b_i(j) & \text{si } j > k, \end{cases}$$

alors $b_i(j) = \lfloor \frac{i}{2^j} \rfloor$ pour tout i,j.

Exercice 5 L'algorithme d'Al Kwarizmi pour multiplier deux entiers opère ainsi:
Diviser le premier entier par 2 en arrondissant à l'inférieur et multiplier le second par 2 jusqu'à ce que le premier vaille 1, puis additionner les valeurs prises par le second lorsque le premier avait une valeur impaire.
Par exemple:

premier	11	5	2	1
second	13	26	52	104

On obtient $11 \times 13 = 13 + 26 + 104 = 143$.

1. Donnez le code Java de cet algorithme.
2. Prouvez sa validité.