Corrigé type

Correction 1. (05pts)

R 1. int recherche(x,deb,fin)

```
{ if(deb==fin)
    {if (x[deb]>=a)&&(x[fin]<=b)     return 1;   ……..(2pts)
     else       return 0;  }
  else {   milieu=(deb+fin)/2;
        int val1= recherche(x,deb, milieu);
        int val2= recherche(x, milieu+1, fin);
         return (val1+val2); }    }
```

R2. non terminale (non simple) **……..(1pt)**

R 3. T(n)= 2+2T (n/2) **……..(1pt)**

R 4. impossible car n'est pas simple **……..(1pt)**

Correction 2. (4pts)

Toutes les réponses avec justification correcte sont acceptées:

1. OUI **……..(1pt)**
2. OUI **……..(1pt)**
3. OUI **……..(1pt)**
4. NON **……..(1pt)**

Correction 3. (6pts)

1. ……..(1pt+1pt)

1. $\Theta(n)$ avec environ n/3 appels récursifs	2. $\Theta(2^{n/2})$. L'arbre des appels est un arbre binaire complet d'hauteur environ n/2. L'´equation de récurrence pour B(n), le nombre d'appels à A lors de l'exécution de A(n) est: $B(n) = 2*B(n-2) +1$ si $n > 1$, $B(n) = 0$ si $n \leq 1$. On obtient $B(n) = 2^{n/2+1}) -1$.

2. voir le cours **……..(2pts)**

3. ……..(2pts)

Dans les deux cas, on peut remarquer que l'instruction la plus souvent exécutée (terme dominant) est "$r \leftarrow r + 1$" avec r initialisé à 0 et retourné par l'algorithme.

Donc la complexité de ces algorithmes est du même ordre de grandeur que la valeur retournée.

— L'algorithme f_1 retourne $\sum_{i=1}^{n}\sum_{j=i+1}^{n} 1 = \sum_{i=1}^{n}(n-i) = n^2 - \sum_{i=1}^{n} i = n^2 - \frac{1}{2}n(n+1) = \frac{1}{2}n(n-1)$: la somme des $n-1$ premiers entiers positifs

Correction 4. (5pts)

1. Il faut distinguer la complexité du calcul de la fonction f, de celle de la fonction elle-même. La complexité de f est en $\mathbf{O(2^n)}$ et en $\mathbf{\Omega(n)}$. Quand à la complexité du calcul de f, il faudrait d'abord écrire un programme qui calcule f et ensuite l'analyser. **……..(0.75pt+0.75pt)**

2. Les équations de récurrence :

D'abord résoudre directement les équations de récurrence de H, et l'on obtient.

Toutes les réponses avec justification correcte sont acceptées:

$H(n) = \frac{1}{2}.n(n+1)$ **……..(1pt)**

complexité est en $\Theta(n^2)$ **……..(0.5pt)**

3. Analyse du petit programme :

On remarque qu'à chaque passage dans la boucle a est divisé par deux et qu'on ajoute 2 à j. **…..(0.5pt)**

La valeur finale de j est donc **2logn**. **…..(0.5pt)**

La complexité de ce programme est donc en **O(logn). …..(1pt)**