Universidad de Costa Rica
Escuela de Ciencias de la
Computación e Informática

CI-1221 Estructuras de Datos
y Análisis de Algoritmos

Práctica No. 1

1.Encuentre el tiempo y orden los siguientes algoritmos:
a.

```
AA(N) {
  A = 2 + BB(N)
  Si CC(N) = DD(N)  {
    Para I = 1  hasta N/2  {
      A = A + I*EE(N,I)
      Para J = I  hasta N  {
        A = A*J*I + FF(N,J)
      }
    }
  }
  Si no {
    A =  GG(N)
  }
  Retornar (A)
}
```

tal que

```
BB(N) {
Si N = 0 o N = 1 {
  Retornar ( 2 )
}
Si no {
  Para I = 1 hasta N  {
    A = A + (N-I)*A + HH(n)
  }
  Retornar ( A*BB(N-2)  –  3*BB(N-1))
 }
}
```

CC(N) tiene un orden de duración igual al del MergeSort

DD(N) tiene un tiempo de ejecución $T(N) = K + 2^N + 2T(N-1)$, $T(1) = K$.

EE(N) tiene un tiempo de ejecución $T(N) = K + N^2 + 5T(N/2)$, $T(1) = K$.

FF(N) es $O(2^N)$

GG(N) tiene un tiempo de ejecución $T(N) = K + N + T(N-2)$, $T(0) = T(1) = K$

HH(N) es $O(N)$

b.

```
{
Leer(n)
X = AA(n)
Si BB(n) es par {
  Para I = 1 hasta n {
    X = CC(n)/DD(n)*I
    Imprimir(X)
    Si X = EE(n) {
      FF(n)
    }
    Sino {
      GG(n)
    }
  }
}
Si no {
  Imprimir(HH(n))
}
}
```

Tal que:

AA(n) tiene un tiempo de ejecución $T(n) = K + n + 3T(n/2)$, $T(1) = K$

BB(n) tiene un tiempo $T(n) = 2T(n-1) + 2T(n-2) + 3n + K$, $T(0) = T(1) = K$

CC(n) tiene un tiempo $T(n) = T(n-1) + 2T(n-2) + n + 2^n + K$, $T(0) = T(1) = K$

DD(n) tiene un tiempo $T(n) = 3T(n-1) + 3^n + K$, $T(1) = K$

EE(n) tiene un tiempo $T(n) = K + n + 3T(n/3)$, $T(1) = K$

FF(n) tiene un tiempo $T(n) = K + n + n^2 + 4T(n/3)$, $T(1) = K$

GG(n) tiene un tiempo $T(n) = K + T(n/3)$, $T(1) = K$

HH(n) tiene un tiempo $T(n) = K + n + n2^n + T(n-1)$, $T(1) = K$

2.Si $O_{A1}(n)$ y $O_{A2}(\log n)$, entonces, podemos afirmar que A2 es mejor que A1? Justifique ampliamente su respuesta.

3.Dado un problema P y un algoritmo A que lo resuelve, se ha determinado que en un tiempo T_0 es posible resolver P para un tamaño máximo N_0. Cuál sería el tamaño máximo del problema si T_0 se duplicara y el orden de duración de A fuera: i. $O(\log n)$, ii. $O(n)$, iii. $O(n^2)$, iv. $O(2^n)$

4.Desde el punto de vista de la complejidad espacio-tiempo compare las siguientes implementaciones de algoritmos:

a.Calcular N! recursivamente e iterativamente.

b.Ordenar una lista de números mediante el algoritmos de Burbuja y hacerlo mediante el MergeSort.

c.Calcular Fibonacci de recursivamente e iterativamente.

d.Búsqueda Secuencial y Búsqueda Binaria.

5.Implementar en C++ el algoritmo para calcular el n-ésimo número de la sucesión de Fibonacci usando el algoritmo recursivo y usando el algoritmo iterativo, en cada caso calcule el número de sumas y el tiempo real de ejecución para hacer una comparación real de ambos algoritmos. Considere también el gasto de espacio. En un cuadro muestre los resultados obtenidos para diferentes valores de n.

6.Calcule tiempos reales de ejecución del algoritmo para resolver el problema de las Torres de Hanoi. En un cuadro muestre los resultados obtenidos para diferentes valores de n.

7.Compare usando tiempos reales de ejecución, los algoritmos de Búsqueda Secuencial (iterativa) y Búsqueda Binaria (recursiva) para el peor caso (el elemento buscado no está en la lista) y para diferentes tamaños de n.

8.Para diferentes tamaños de n (pequeños, medianos y grandes) haga un cuadro que muestre los diferentes valores para las funciones : log n, n, n log n, n^2, 2^n y haga la conversión a tiempos reales: segundos, minutos, días, años, etc., con el fin de tener una idea de las diferencias reales en términos del tiempo de ejecución de algoritmos cuyos órdenes de duración son iguales a las funciones dadas.