Tarea 1

Fundamentos de Estructuras de Datos y Algoritmos

Primer Semestre 2019, Prof. Cecilia Hernández

1. Ejercicios

1. [0.5 puntos] Una función $f(n)$ es $O(g(n))$ si existen constantes $n_0 \geq 0$ y $c > 0$ tal que $f(n) \leq c \times g(n)$ para $n \geq n_0$. Proporcione un análisis $O()$ para cada una de las siguientes funciones. En cada caso, muestre los valores de las constantes n_0 y c que hacen cierta su afirmación.

   a) $n^2 + \sqrt{n} - 2$

   b) $5\log_3(\log(\log(n)/20))$

   c) $5n + 3\log(n^2)$

   d) Pruebe formalmente que $O()$ es una relación transitiva. Esto es si $f(n) \leq O(g(n))$ y $g(n) \leq O(h(n))$, entonces $f(n) \leq O(h(n))$

2. [1 puntos] Determine si las siguientes afirmaciones son verdaderas o falsas. Justifique su respuesta.

   a) $log(n!)$ es $O(log(n))$

   b) $4n^3$ es $\omega(n^3)$

   c) $(\sqrt{2})^{\log(n)}$ es $\theta(\sqrt{n}))$

   d) $\sqrt{n} - 3\log(n^{10})$ es $\Omega(n)$

3. [1 punto] Determine qué realiza el siguiente segmento de código y pruebe su correctitud.

```
int y=0, i=0;
while ( i <= n ) {
    y += 2^i;
    i++;
}
cout<<"y = "<< y <<endl;
```

4. [1 punto] Proporcione un análisis asintótico de peor caso en notación $O()$ para el tiempo de ejecución de los siguientes fragmentos de programa.

(a)

```
int x = 0;
for (int i = 0; i < n; i++) {
  for (int j = 0;j <n*n/3; j++) {
         x += j;
  }
}
```

(b)

```
int x = 0;
if (n % 3 == 0) {
  i=0;
  while(i < n*n*n)
     x++;
     i++;
  }
} else {
  for(int i=0; i<n*n*n; i++) {
     x++;
  }
}
```

(c)

```
int recurse(int n) {
   for (i = 0; i < n*n; i += 2) {
      procesar(i); // O(1)
   }
   if (n <= 0)
      return 1;
   else
      return recurse(n-3);
}
```

(d)

```
int i=n;
int j,k;
while (i > 1){
   j = i;
   while(j < n){
      k = 0;
      while (k < n){
         k = k + 2;
      }
      j = j*2;
   }
   i = i/2;
}
```

5. [2.5 puntos] Considere un arreglo ordenado de números enteros A de tamaño n y un valor entero x ingresado por teclado. Se desea encontrar las posiciones y valores de un par de elementos $(A[i], A[j])$ en el arreglo A tal que la suma $A[i] + A[j] = x$, si es que existen. Solo necesita reportar el primer par encontrado. En caso de no encontrarse tal par de elementos, el algoritmo debe retornar que no existen.

   Ejemplos:

   $A = [-5, -2, 0, 1, 3, 3, 5, 5, 8, 8, 9, 12]$, $x = 10$,
   los posibles pares son:
   $[-2, 12]$ en pos $(1, 11)$, $[1, 9]$ en pos $(3, 10)$, $[5, 5]$, $(6, 7)$

   $A = [1, 3, 3, 5, 5, 8, 9, 12]$, $x = 10$,
   los posibles pares son:
   $[1, 9]$ en pos $(0, 6)$, $[5, 5]$, $(3, 4)$

   $A = [1, 2, 2, 5, 6, 8, 12, 12]$, $x = 11$,
   los posibles pares son:
   $[5, 6]$ en pos $(3, 4)$

a) Escriba un pseudo código para un algoritmo $O(n^2)$ que resuelva el problema.

b) Ahora diseñe un algoritmo $O(n\log(n))$ que resuelva el problema y escriba su pseudo código.

c) Ahora diseñe un algoritmo $O(n)$ que resuelva el problema y escriba su pseudo código.

d) Implemente los tres algoritmos usando C++ definiendo una función para cada algoritmo. Además considere dos tipos de entrada posible. Un tipo donde los elementos del arreglo son elegidos aleatoriamente y un tipo de entrada que corresponda al peor caso de cada algoritmo. Note que para el tipo aleatorio debe ordenar el arreglo antes de aplicar los algoritmos que solucionan el problema propuesto. Construya un gráfico que muestre como varía el tiempo de ejecución en nanosegundos variando el tamaño de la entrada (n). Para ello considere diversos valores para n. Puede utilizar potencias de 2, osea $256, 512, 1024, 2048, 4096, 8192$. Solo debe medir el tiempo que corresponde a la ejecución de la función.

6. Medición de tiempos.

```
#include <chrono>

using namespace std;

auto start = chrono::high_resolution_clock::now();
auto finish = chrono::high_resolution_clock::now();
auto d = chrono::duration_cast<chrono::nanoseconds> (finish - start).count();
cout <<"total time "<< duration << " [ns]" << " \n";

// nota, si estima necesario puede usar milliseconds
// en lugar de nanoseconds
```

Fecha de Entrega: Lunes 22 de Abril 11:59PM