TC2017 – Análisis y Diseño de Algoritmos ***Tarea #3***
Ing. Luis Humberto González G *Fecha de Entrega: 27-Ago-2013*
Nombre: *Matricula:*

Entrega un reporte de solución por escrito a cada uno de los siguientes problemas.

1. *(40 puntos)* Analizar cada uno de los siguientes segmentos de pseudocódigo, e indicar cuál es la complejidad del algoritmo representado. Suponer siempre que la "instrucción" es una operación básica.

CASO A:

```
for (j = 1; j <= n ; j = j * 2)
   instrucción;
```

CASO B:

```
for (j = 1; j <= n ; j++)
  for (k = 0; k < n/5; k++)
     instrucción;
```

CASO C:

```
for  (j = 1; j <= n ; j++)
{  k = n;
   while (k >= 1) do
    {  instrucción;
       k = k / 2;
    }
}
```

CASO D:

```
 j = n;
while j > 0  do
   {  for ( k = j; k<=n; k = k*2)
          instrucción;
      j = j / 2;
   }
```

2. *(20 puntos)* A continuación se muestran 2 casos en los que para cada uno se muestran 2 algoritmos. Ambos algoritmos en cada caso, sirven para resolver el mismo problema.
   a) Identifica cuál es el problema que está resolviéndose en cada caso.
   b) Haz un análisis de la complejidad de cada algoritmo.
   c) Indica para cada caso, cuál es el algoritmo que más convendría elegir para la implementación de la solución al problema, justificando tu respuesta.

CASO A:

Algoritmo A1:

```
s = 0;
for x = 1 to n do
  for y = 1 to n do
     if x = y then
          s = s + a[x,y];
```

Algoritmo A2:

```
s = 0;
for j = 1 to n do
   s = s + a[j,j];
```

CASO B:

Algoritmo B1:

```
r = 1;
fo(int  j=1; j<=n; j++)
   r = r *  x
```

Algoritmo B2:

```
r = 1;
   while (n > 0)
   {  if  (n % 2 <>  0)  r = r * x;
     x = x * x;
    n = n / 2;
   }
```

3. *(40 puntos)* Desarrolla un algoritmo que dado un arrelo **A** que contiene **n** distintos enteros positivos, generé un arreglo de dos dimensiones **B**, en donde en la posición **B[i][j] = B[j][i] = A[i]+A[i+1]+…+A[j-1]+A[j].** Se calificará eficiencia.

Análisis y Diseño de Algoritmos ***Tarea #4***
Ing. Luis Humberto González G *Fecha de Entrega: 3 Sep 2013*
Nombre: *Matricula:*

1) *(10 puntos) Contesta las preguntas en base al* siguiente algoritmo

```
s = 0
for (int i=1; i<=n; i++)
  s = s + i * i
return s
```

a) ¿Qué realiza el algoritmo? _____

b) ¿Cuál es la operación básica? _____

c) ¿Cuántas veces se realiza la operación básica? _____

d) ¿Cuál es la complejidad del algoritmo? _____

e) ¿Cuál es el orden del algoritmo? _____

2) *(40 puntos)* ¿Cuál es el orden de cada uno de los siguentes algoritmos?

a) // Entrada: Matriz A[0..n-1, 0..n-1] de números reales.

```
for (int i=0; i<= n-2; i++)
  for (int j=i+1; j<n; j++)
    for (int k=I; i<=n; i++)
      A[i,k] = A[j,k] –A[i,k] * A[j,i] / A[i,i]
```

_____

b) //Entrada: Un entero positivo (n)

```
int Q(int n){
  if (n==1) ´
    return 1
  return Q(n-1)+2*n-1
}
```

_____

c) //Entrada: Un entero positivo (n)

```
int P(int n){
  if (n==0)
     return 0
  else
    if (n % 2 == 0)
       return n+P(n-3)
    else
       return n+P(n-1)
}
```

_____

d) //Entrada: Un entero positivo (n)

```
int a=0;
int b=n;
for (int i=1; i<= 2*n; i++) {
            a++;
            b+=a;
            c*=(a+b);
}
b=c+a;
```

_____

e) //Entrada: Un entero positivo (n)

```
int acum=1;
for (int i=1; i<=n; i++)
   for (int j=i;j<=n; j++)
      acum+=(i*j);
```

_____

f) //Entrada: Un entero positivo (n)

```
int b=1;
j = n;
while (j>=0) {
    b++;
    j--;
}
```

_____

g) //Entrada: Un entero positivo (n)

```
int acum=1;
for (int i=1; i<=n; i+=2)
   for (int j=i;j<=n; j++)
      acum+=(i*j);
```

_____

h) //Entrada: Un entero positivo (n)

```
int acum=1;
for (int i=1; i<=n; i*=2)
   for (int j=i;j<=n; j+=2)
      acum+=(i*j);
```

_____

3) *(50 puntos)* Escribe un algoritmo que dado un arreglo que contiene enteros positivos, regrese la suma de los enteros impares contenidos en el arreglo.

   a) Realiza el algoritmo en forma iterativa, ¿Cuál es el orden del algoritmo?

   b) Realiza el algoritmos en forma recursiva ¿Cuál es el orden del algoritmo?

Análisis y Diseño de Algoritmos ***Tarea #5***
Ing. Luis Humberto González G *Fecha de Entrega: 3 Sep 2013*
Nombre: *Matricula:*

1) *(50 puntos)* Soluciona las siguientes ecuaciones recursivas, llegando a su forma cerrada.

   a. $T(n) = T(n/2)+1 \quad n>1; \; T(1) = 1$ _____
   b. $T(n) = 3T(n-1) \quad n>1; \quad T(1) = 4$ _____
   c. $T(n) = Q(n/2) + n \quad n>1; \; T(1) = 1$ _____
   d. $T(n) = 3T(n/4)+2 \quad n>1; \; T(1) = 2$ _____
   e. $T(n) = T(n-2)+1 \quad n>2; \quad T(2) = 1$ _____

2) Escribe un algoritmo recursivo que dado una matriz cuadrada de ***nxn***, que contiene enteros positivos, regrese la cantidad de casillas con valor mayor a 100,

   a) *(30 puntos)* Realiza el algoritmo recursivo.
   b) (10 puntos) ¿Cuál sería la formula recursiva del tiempo de ejecución?
   c) (10 puntos) Encuentra la solución de la formula recursiva del inciso b.

TC2017 – Análisis y Diseño de Algoritmos ***Tarea #6***
Ing. Luis Humberto González G *Fecha de Entrega: 17-Sep-2013*
Nombre: *Matricula:*

Entrega un reporte de solución por escrito a cada uno de los siguientes problemas.

1. El algoritmo del ***Bubble Sort*** es muy popular por su simplicidad. Si lo recuerdas, la idea de este algoritmo consiste en recorrer todo el arreglo de datos, comparando cada elemento con su sucesor, y en caso de que estos 2 datos estén desordenados, intercambiarlos. El final del recorrido de todo el arreglo, se asegura que el elemento más grande (si se está ordenando ascendentemente) ha quedado en la última posición. Por lo tanto, el algoritmo tiene que recorrer de nuevo el arreglo desde el inicio con el mismo proceso para dejar el siguiente elemento mayor en la penúltima posición. Obviamente, este proceso se repite hasta que se ordenan los últimos 2 datos en el arreglo.
   a) *(10 puntos)* En base a esta descripción del Bubble Sort, escribe en pseudocódigo (o en el lenguaje de tu preferencia) el algoritmo computacional que describe a este proceso.
   b) *(5 puntos)* Explica si este algoritmo se puede analizar obteniendo el comportamiento para el peor caso, el caso promedio y el mejor caso, o bien, si es un algoritmo que tiene un comportamiento igual para todos los casos ("every case"). Justifica.
   c) *(10 puntos)* Con las herramientas de análisis general que hemos visto en clase, obtén el orden de este algoritmo. Trata de justificar lo mejor posible tu respuesta.
   d) *(10 puntos)* Aplica el algoritmo sobre un arreglo que ya está ordenado. ¿Qué observas? Como te darás cuenta, el algoritmo trabaja sin sentido para ordenar algo que ya está ordenado. Puesto que esto es crítico, un Bubble Sort mejorado podría considerar que cuando no hay intercambios en el barrido del arreglo, éste ya está ordenado, y por lo tanto se puede terminar el proceso de ordenamiento. Modifica el algoritmo que escribiste anteriormente, para que en una nueva versión se considere esta mejora.
   e) *(5 puntos)* De nuevo responde los incisos c) y d) pero ahora para la nueva versión del algoritmo que obtuviste en el inciso e).

2.
   a) *(10 puntos)* Escribe un algoritmo eficiente para buscar un dato en una matriz M de n X n. La matriz contiene a los datos ordenados por renglones de tal manera que:

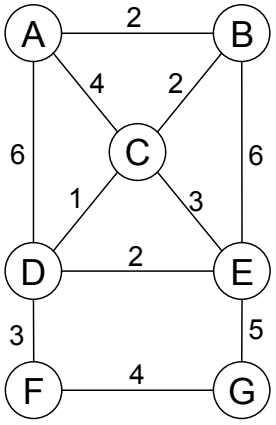
$$M[i,j] < M[i,j+1] \ , M[i,j] < M[i+1, j] \ y \ \mathbf{M[i,n] < M[i+1, 1]}$$

   El algoritmo deberá tener al menos un comportamiento de orden lineal y no cuadrático.
   b) *(5 puntos)* Para una cantidad de n2 datos ordenados, explica si este algoritmo pudiera resultar mejor que la búsqueda binaria en el peor caso. Justifica tu respuesta.

3. Dada la siguiente lista de datos: **87 23 12 15 76 91 44 56**
   a) *(7 puntos)* Muestra, paso a paso, la forma en que se hacen las modificaciones a la lista si se aplica el algoritmo del Merge Sort para ordenarla descendentemente (de mayor a menor).
   b) *(8 puntos)* Muestra, paso a paso, la forma en que se hacen las modificaciones a la lista si se aplica el algoritmo del Quick Sort para ordenarla descendentemente (de mayor a menor) considerando como elemento pivote al último elemento en la lista.
   c) *(10 puntos)* Para este caso particular, compara el comportamiento de ambos algoritmos, contando la cantidad de veces que se ejecuta la operación básica, que en este caso es la comparación de datos en los subalgoritmos Une y Partición presentados en clase. ¿Cuál algoritmo resultó ser mejor para esté caso?
   d) *(5 puntos)* Menciona un ejemplo de una secuencia de 8 datos en que el algoritmo del Merge Sort se comporte mejor que el Quick Sort para ordenarlos ascendentemente.

4. *(15 puntos)* Utilizando la técnica de divide y vencerás plantea un algoritmo que sirva para encontrar el elemento menor de un arreglo de datos desordenados. Escribe el algoritmo computacional en pseudocódigo. Haz un análisis general y explica si el planteamiento de solución con la técnica de divide y vencerás representa alguna ventaja con respecto al algoritmo que en forma tradicional se emplearía para realizar este proceso. Justifica tu respuesta.

1. *(20 puntos) R*esuelve paso por paso el problema de obtener los caminos más cortos para el siguiente grafo: (A=1, B=2, C=3, D=4, E=5, F=6, G=7)



   Como resultado de este ejercicio deberás entregar lo siguiente
   a) Como quedaría las matrices D^3 y D^7 que genera el algoritmo de Floyd.
   b) Como queraría la matriz D^7.

2. *(20 puntos)* Resuelve la secuencia de dimensiones d_0 a d_4 de las matrices que se van a multiplicar en forma encadenada, y que utilizando el algoritmo de Godbole, obtenga como resultado el número mínimo (óptimo) de multiplicaciones escalares que se realizarían.

   Como resultado de este ejercicio deberás entregar lo siguiente:
   a) Realiza el ejercicio con los siguientes valores de dimensiones de matrices:

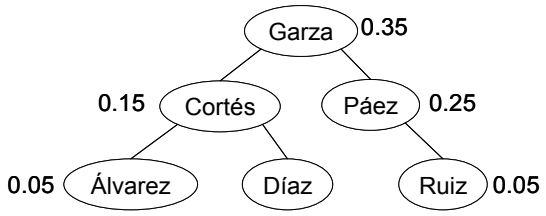
d_0	d_1	d_2	d_3	d_4
3	12	15	2	10

   b) Obtener a la matriz P que se utilizaría para construir la secuencia de multiplicaciones, e indicas cuál es el resultado para la prueba del inciso a.

3. *(10 puntos)* ¿Cuál es el valor que guardará el algoritmo de Godbole en la casilla [1,n] de la matriz de resultados **D**, si se multiplican ***n*** matrices cuadradas de dimensiones ***m*** X ***m*** ? Expresar el resultado en términos de las variables ***m*** y ***n***. Justifica tu respuesta.

4. *(10 puntos)* Si se aplica el algoritmo para obtener el ABB óptimo sobre 8 palabras que tienen exactamente la misma probabilidad de ser buscadas, ¿cuál sería el valor de la casilla [1,8] de la matriz **A** que genera el algoritmo? Justifica tu respuesta.

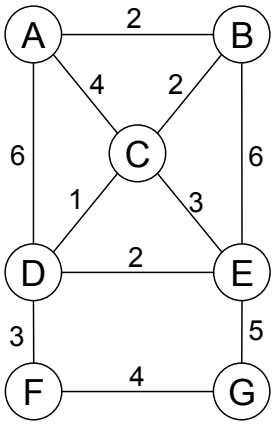
5. *(20 puntos)* El siguiente árbol fue construido siguiendo el algoritmo para obtener al ABB óptimo. Los valores que acompañan a *algunos* nodos, corresponden a la probabilidad de búsqueda asignada a esas llaves.



Responde a los siguientes incisos:

a) ¿Cuál es el valor de la casilla [4, 6] en la matriz que calcula el mínimo promedio de comparaciones en el árbol?
b) ¿Cuál es el valor de la casilla [2, 3] en la matriz que calcula el mínimo promedio de comparaciones en el árbol?
c) ¿Cuál es el mínimo promedio de comparaciones en el árbol y en qué posición de la matriz se encuentra este resultado?
d) ¿Cuál es la matriz de raíces R de donde se construyó este árbol?

6. *(20 puntos)* Para el siguiente grafo aplica el problema del viajero y contesta las siguientes preguntas



a) ¿Cuántos ciclos hamiltonianos existen y cuales sonl?
b) ¿Qué almacena la casitlla [D, { B, E } ?
c) ¿Qué almacena la casilla [B, { D, F, E, G }?
d) ¿Cuál es la la ruta optima y cuál es su costo?

Análisis y Diseño de Algoritmos ***Tarea #13***
Ing. Luis Humberto González G *Fecha de Entrega: 21 de Noviembre 2013*

1. *(25 puntos)* En el problema de 5 reinas con tablero 5x5
   a) ¿Cuál es la última solución encontrada en el problema de 5 reinas (tablero 5x5)?
   b) Explique como la simetría del tablero puede ser usada para localizar otras soluciones.

2. *(25 puntos)*Los siguientes objetos se desean colocar en un recipiente que tiene capacidad para soportar hasta ***10*** unidades de peso:
   - OBJETO A: vale ***$50*** y pesa ***5***.
   - OBJETO B: vale ***$18*** y pesa ***3***.
   - OBJETO C: vale ***$66*** y pesa ***6***.
   - OBJETO D: vale ***$35*** y pesa ***5***.

   Si se deseara llenar el recipiente con un subconjunto de objetos, de tal manera que se maximice el valor guardado sin exceder la capacidad del recipiente. Escribe el árbol de decisión que se generaría con la técnica de backtracking.

3. *(25 puntos)* Se tienen 6 objetos con los siguientes pesos: ***1, 3, 4, 5, 6, 7*** y se desea obtener los conjuntos de objetos que acumulan un peso exacto de ***10***.
   a) Dibuja el árbol de búsqueda de soluciones usando backtracking.
   b) Señala todas las soluciones encontradas.
   c) ¿Cuantos nodos con valor de cero se tienen usando backtracking?
   d) ¿Cuántos nodos en total genera el árbol de búsqueda de soluciones usando backtracking?

4. *(25 puntos)* Considera el siguiente problema:

   > "Se sabe que para asignar los equipos que trabajaran para el proyecto final de la materia, el maestro esta solicitando que en el equipo no este integrado por personas que son de la misma ciudad, ni que sean novios. Se sabe que Lety, Emilio y José son de Cd. Acuña, que Amanda y Pablo de Nogales, además de que Alejandra y Carlos son de Piedras Negras. Por otro lado Lety es novia de Pablo, Emilio es novio de Alejandra y Amanda es novia de Carlos."

   Dibuja el árbol de decisión que genera el algoritmo de backtracking hasta encontrar la primera solución.

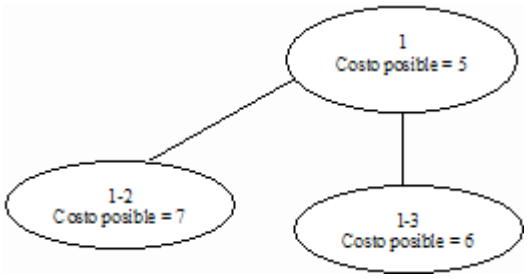
1. Los siguientes objetos, se desean colocar en un recipiente que tiene capacidad para soportar hasta 20 unidades de peso:

- OBJETO 1, vale $9 y pesa 3.
- OBJETO 2, vale $20 y pesa 4.
- OBJETO 3, vale $6 y pesa 3.
- OBJETO 4, vale $12 y pesa 2.
- OBJETO 5, vale $16 y pesa 4.
- OBJETO 6, vale $8 y pesa 8.

Si se deseara llenar el recipiente con un subconjunto de objetos, de tal manera que se maximice el valor guardado sin exceder la capacidad del recipiente, utilizando la técnica de Branch and bound para encontrar la solución, responde a los siguientes incisos:

a) *(5 puntos)* ¿Cuántos niveles podría llegar a tener el árbol?
b) *(5 puntos)* ¿Cuáles son los objetos que posiblemente estén acumulados en un nodo del nivel 3 del árbol de búsqueda de soluciones?
c) *(5 puntos)* ¿Cuál es el valor posible a acumular al inicio del algoritmo (nodo raíz del árbol de búsqueda de soluciones)?
d) *(5 puntos)* ¿Cuál es el peso acumulado en el tercer nodo que se genera en el árbol de búsqueda de soluciones por el algoritmo de backtracking?
e) *(5 puntos)* ¿Cuál es el valor real acumulado en el tercer nodo que se genera en el árbol de búsqueda de soluciones por el algoritmo de Branch and bound?
f) *(25 puntos)* Realiza el árbol completo.

2. El siguiente es el árbol de búsqueda de soluciones para el problema del viajero por la técnica de Branch and Bound, después de haber generado los primeros 3 nodos:



Si la matriz de adyacencias del grafo de donde se ha obtenido este árbol es la siguiente:

$$\begin{pmatrix} 0 & 1 & 2 & 3 \\ 1 & 0 & 1 & 2 \\ 2 & 1 & 0 & 3 \\ 3 & 2 & 3 & 0 \end{pmatrix}$$

Responde a los siguientes incisos:

a) *(5 puntos)* ¿Cuál es el costo mínimo óptimo que considera el algoritmo para hacer comparaciones y tomar decisiones (en el estado en que se encuentra el árbol)?
b) *(5 puntos)* Dibuja en el árbol, el siguiente nodo a analizar e indica claramente sus datos y si este nodo es candidato a expandirse o no.
c) *(5 puntos)* Dadas las condiciones del grafo, y la información que puede dar el árbol de búsqueda de soluciones completo, ¿cuántos ciclos hamiltonianos tiene el grafo?
d) *(10 puntos)* ¿Cuál es el primer ciclo hamiltoniano que daría el algoritmo de backtracking con este grafo?
e) *(10 puntos)* ¿Cuál es el primer ciclo hamiltoniano que el algoritmo de Branch and bound encuentra como solución al problema del viajero con este grafo?
f) *(15 puntos)* Completa el árbol.