Programación 2 UNMDP
Totalizador - Agosto 2018

1 - Indique V o F y justificar, cualquier respuesta no justificada obtendrá 0 puntos:
a) Se puede obtener una matriz de transición, haciendo un ciclo a través de una matriz proveniente del algoritmo de Floyd.
b) Se puede implementar una pila estática que en vez de arrancar en 0 arranque en N.

2 - Dado un árbol binario proveniente de un bosque, calcular el nivel mayor.

3 - (Usar TDA N-Ario) Generar una lista doble en la cual cada nodo posee la suma de los elementos ubicados en niveles pares con cantidad de elementos pares de un árbol N-ario.
* No usar variables auxiliares.

4 - Generar una cola circular que en cada nodo contenga clave, y el grado de entrada de un grafo dirigido de N vértices que se encuentra en una lista de adyacencia.
* No utilizar estructuras auxiliares.
* Implementar los operadores utilizados de la TDA Cola circular.

Para cada ejercicio mostrar su invocación.

Serán considerados al calificar este examen la eficiencia de las soluciones y el uso de las características del lenguaje C y la programación estructurada.
*La nota final de la materia se calculará: 0,3*nota cursada + 0,7*nota totalizador*

Agosto 2018

1- V o F

a) Se puede obtener una matriz de transicion haciendo a través de una matriz proveniente del algoritmo de Floyd VERDADERO

La matriz de Floyd tiene la información suficiente para construir a partir de ella una matriz de alcance.

b) Se puede implementar una pila estática que en vez de arrar en 0 lo haga en N. VERDADERO

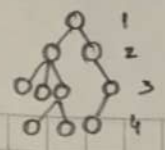
Se puede decrementando el tope al poner un elemento e incrementandolo al momento de poner.

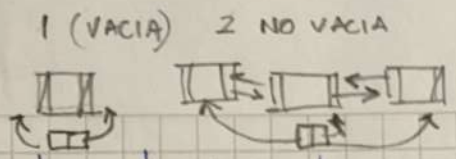
2- Dado un arbol binario proveniente de un bosque calcular el nivel mayor

```
int MayorNivel(arbol a){
  int nivelizq, nivelder, nivelmax;
  if (a!=NULL){
    nivelizq = 1+ MayorNivel(a->izq);
    nivelder = MayorNivel(a->der);
    if (nivelder > nivelizq)
      nivelmax = 1+ nivelder;
    else
      nivelmax = 1+ nivelizq;
  }
  else
    nivelmax = 0;
  return nivelmax;
}
int main(){
  arbol bosque = NULL;

  Carga(&bosque);

  Printf("Maximo nivel: %d", MayorNivel(bosque));
  return 0;
}
```





3- Generar una lista doble en la cual cada nodo posee la suma de los elementos ubicados en niveles pares con cantidad de elementos pares de un arbol N-ario

```
int SumaNivPares(ArbolN A, posicion c, int nivel){
    Posicion c;
    int suma, Acum=0, CantElem;
    if(!Nulo(P)){
        c=HijoMasIzq(P,A);
        Suma=0;
        CantElem=0;
        While(!nulo(c)){
            if(nivel%2==0){
                Suma+=Info(C,A);
                CantElem++;
            }
            Acum+=SumaNivPares(A,C,nivel+1);
            C=HnoDer(C,A);
        }
        if(CantElem%2==0)
            Acum+=Suma;
    }
    return Acum;
}
void Generalista(TListaD *LD, ArbolN A){
    PnodoD nuevo;
    nuevo=(PnodoD) malloc(sizeof(nodoD));
    nuevo->dato=SumaNivPares(A, Raiz(A), 1);
    if((*LD).pri==NULL){
        nuevo->ant=NULL;
        (*LD).pri=nuevo;
    }
    else{
        (*LD).ult->sig=nuevo
        nuevo->ant=(*LD).ult
    }
    nuevo->sig=NULL
    (*LD).ult=nuevo;
}
```

4- Generar una cola circular que en cada nodo contenga clave, y el grado de entrada de un digrafo dirigido de N vertices que se encuentra en una lista de adyacencia

```
void GeneralC(TListaAd LAd[], int N, TCola *C){
    TElementoC RegV; IniciaC(C);
    TListaAd aux;
    int gradoE, i, j;
    for (i=0; i<N; i++){
        gradoE = 0;
        for (j=0; j<N; j++){
            aux = LAd[j];
            while (aux!=NULL){
                gradoE += aux->vertice == j+1;
                aux = aux->sig;
            }
        }
        RegV.Clave = i+1;
        RegV.GE = gradoE;
        PonC(C, RegV);
    }
}
```