UNL

MI CALENDARIO

MI CUENTA

Página Principal ► Mis cursos ► AED 2021 ► Evaluaciones ► TPL 2 2021 Pregunta 1 Navegación por el cuestionario Sin responder aún

3 Terminar intento... Tiempo restante 0:46:01 Puntúa como 1,00 Marcar Marcar pregunta

VIDEO EXPLICATIVO: O 👌 e-fich.unl.edu.ar/mo New File Open Open Directory Close Save Undo Cut Copy Paste Search Download Vista previa de la pre > garbag 3 clase-2 ARBAGE/ZOOM clase-2 **ADMINISTRACION** prac.m RBAGE/ZOOM RBAGE/ZOOM applym RBAGE/ZOOM Desktop partic. mstorti-scratch doit.sh RBAGE/ZOOM Ø doit.sh. ARBAGE/ZOOM T=(9 (1 2 3)) => M={0->2,1->1,2->1} T=<mark>(</mark>1 (3 3 2) (2 3) (2 (1 (2 3))) (1 1)) => M={0->5,1->4,2->1,3->1,4->1}, ARRAGE/700M Ø file.txt. Ø doit.sh. ARBAGE/ZOOM Images U:**- *scratch* All L3 (Lisp Interaction) 4:02pm 2.00 clase-2021-10-23_15.59.59.mov (QuickTime vio 172.2 GiB free

de altura m. Ejemplo: T=(9 (1 2 3))=> M={0->2,1->1,2->1} $T=(1 (3 3 2) (2 3) (2 (1 (2 3))) (1 1)) => M={0->5,1->4,2->1,3->1,4->1},$

Escribir una función void countheight(tree<int> &T,map<int,int> &count); que devuelve en count el conteo de nodos de un árbol T por altura, es decir count[m] debe contener la cantidad de nodos

AYUDA:

 Recordar que la altura de un nodo se puede calcular como el máximo de la altura de los hijos +1 • Como casos especiales, Las hojas tienen altura 0 y los Lambda tienen -1.

• Escribir una función recursiva sobre los nodos n, que retorna la altura del nodo, y además va actualizando el contador.

Descargar ZIP * En el CPP deben incluir su nombre y DNI así:

Adjuntar el archivo program.cpp con la implementación solicitada renombrándolo

// Nombre: Esponja, Bob. DNI: 23456789

así: esponjabob.cpp (si su nombre es Bob Esponja). El ejercicio será válido solo si pasa satisfactoriamente TODOS los tests automáticos.



Pregunta 2

Marcar Marcar

pregunta

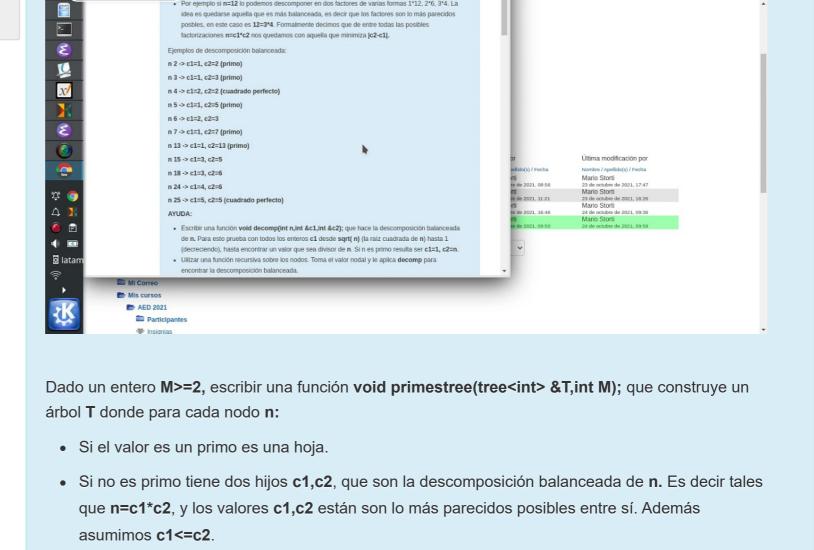
Sin responder aún

Puntúa como 1,00

VIDEO EXPLICATIVO

1<=c2.

10:01



gory=16096%2C73588&lastch... 🛊 🧔 🖸 🍗 🛊 👗 🎅 🗄

lica 🖿 other

Other bookmarks Reading list

Vista previa de la pregunta - Google Chrome edu.ar/moodle/question/preview.php?id=95187&courseid=1145

i no es primo tiene dos hijos c1,c2, que son la descomposición balanceada de n. Es decir tales

e n=c1*c2, y los valores c1,c2 están son lo más parecidos posibles entre sí. Además asumir

• Por ejemplo si **n=12** lo podemos descomponer en dos factores de varias formas 1*12, 2*6, 3*4. La idea es quedarse aquella que es más balanceada, es decir que los factores son lo más parecidos posbles, en este caso es **12=3*4**. Formalmente decimos que de entre todas las

posibles factorizaciones **n=c1*c2** nos quedamos con aquella que minimiza **|c2-c1|**.

- Ejemplos de descomposición balanceada: n 2 -> c1=1, c2=2 (primo) n 3 -> c1=1, c2=3 (primo)
- n 7 -> c1=1, c2=7 (primo) n 13 -> c1=1, c2=13 (primo)

n 4 -> c1=2, c2=2 (cuadrado perfecto)

n 15 -> c1=3, c2=5 n 18 -> c1=3, c2=6 n 24 -> c1=4, c2=6

n 5 -> c1=1, c2=5 (primo)

n 6 -> c1=2, c2=3

EJEMPLOS DE PRIMESTREE: $M=12 \Rightarrow T=(12\ 3\ (4\ 2\ 2))$ $M=9 \Rightarrow T=(9 3 3)$

n 25 -> c1=5, c2=5 (cuadrado perfecto)

AYUDA: • Escribir una función void decomp(int n,int &c1,int &c2); que hace la descomposición balanceada de n. Para esto prueba con todos los enteros c1 desde sqrt(n) (la raíz cuadrada de

función a cada uno de esos nodos.

c1=1, c2=n.

 $M=27 \Rightarrow T=(27 \ 3 \ (9 \ 3 \ 3))$

M=18295200

 Uilizar una función recursiva sobre los nodos. Toma el valor nodal y le aplica decomp para encontrar la descomposición balanceada. • Si c1=1 entonces n es primo y no hay que hacer nada. • Si c1!=1 entonces inserta como hijos de n a c1 y c2 (en ese orden) y aplica recursivamente la

n) hasta 1 (decreciendo), hasta encontrar un valor que sea divisor de **n**. Si n es primo resulta ser

=> T=(18295200 (4235 (55 5 11) (77 7 11)) (4320 (60 (6 2 3) (10 2 5)) (72 (8 2 (4 2 2)) (9 3 3))))

Descargar ZIP * En el CPP deben incluir su nombre y DNI así: // Nombre: Esponja, Bob. DNI: 23456789

Adjuntar el archivo program.cpp con la implementación solicitada renombrándolo

así: esponjabob.cpp (si su nombre es Bob Esponja). El ejercicio será válido solo si

// Nombre: 1 int D

for(int i=sqrt(n);i>=1;--i) { c1=n/i;if(i*c1==n){

if(t.begin()==it)it=t.insert(it,m);

int a=1, b=m; decomp(m,a,b);

if (b==1) return;

auto c=it.lchild();

c=t.insert(c,min(a,b));

void decomp(int n,int &c1,int &c2){

pasa satisfactoriamente TODOS los tests automáticos.

c2=i;return;

void div(tree<int> &t,tree<int>::iterator it, int m) {

```
div(t,c,min(a,b));
  C++;
  c=t.insert(c,max(a,b));
  div(t,c,max(a,b));
void primestree(tree<int> &t,int m) {
 auto it=t.begin();
  div(t,it,m);
                            Tamaño máximo de archivo: 80MB, número máximo de archivos:
Archivos
```

Pregunta 3

Marcar Marcar

pregunta

Sin responder aún

Puntúa como 1,00

* En el CPP deben incluir su nombre y DNI así: // Nombre: Esponja, Bob. DNI: 23456789 Adjuntar el archivo program.cpp con la implementación solicitada renombrándolo así: esponjabob.cpp (si su nombre es Bob Esponja). El ejercicio será válido solo si pasa satisfactoriamente TODOS los tests automáticos.

bool DFD(graph &g, map<int,int> &m, map<int,bool> &v,int nodo1, in

for(auto it=g[nodo1].begin(); it!=g[nodo1].end(); ++it){

if(m[*it]>=enlace && *it==nodo2) return true;

Dado un grafo G map <int, list<int> > que representa la topología de interconexón de una red

velocidades máximas de conexión por cada nodo. Escribir una función bool chk_enlace(graph

&G,map<int,int> &M,int pc1, int pc2, int enlace); donde se reciben además como parámetros

dos nodos con el id de cada pc y la velocidad mínima en mb que debe cumplirse en todo el

enlace entre ambas. En caso de existir tal enlace que cumpla con ese mínimo de velocidad se

• Implementar el algoritmo para verificar un camino existente entre dos nodos de un grafo y

http://e-fich.unl.edu.ar/moodle/pluginfile.php/73588/question/questiontext/141493/3/95428/tpl2-2021-

de computadoras, y una correspondencia M donde se encuentran almacenadas las

retorna true, caso contrario false.

luego agregarle la condición del enlace

AYUDA:

ZIP:

chk_enlace.zip

// Nombre:

v[nodo1]=1;

if(nodo1==nodo2) return true;

int

```
else if(v[*it]==0 && m[*it]>=enlace)return DFD(g,m,v,*it,nodo
  return false;
bool chk enlace ( graph &g, map<int, int > &m, int pc1, int pc2, int e
 bool flag=0; map<int,bool> v;
  flag=DFD(g,m,v,pc1,pc2,enlace);
  return flag;
```

Reiniciar tour para usuario en esta página Reiniciar tour para usuario en esta página