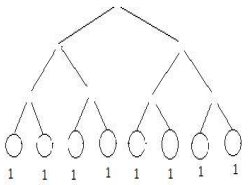
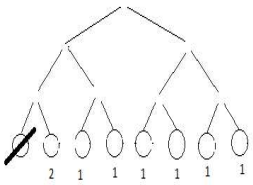
Anexo 1. Caso de N=8 elementos

De forma totalmente análoga al caso de N=4 elementos, podremos ver el caso no trivial de N=8 elementos.

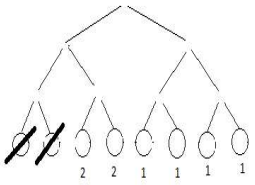
Etapa 1



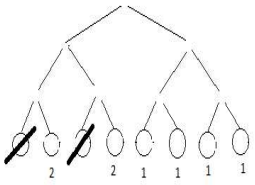
Etapa 2



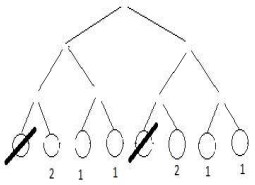
Etapa 3.1



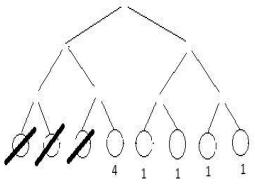
Etapa 3.2



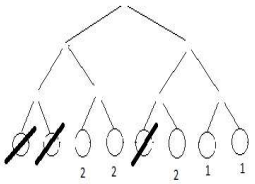
Etapa 3.3



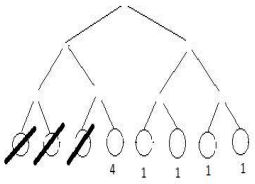
Etapa 4.1



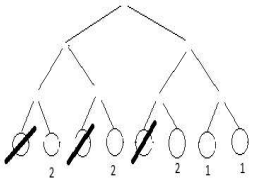
Etapa 4.2



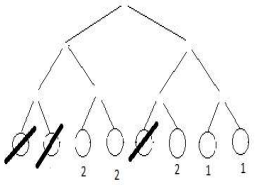
Etapa 4.3



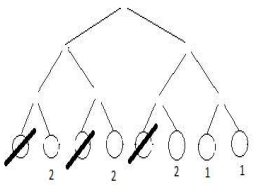
Etapa 4.4

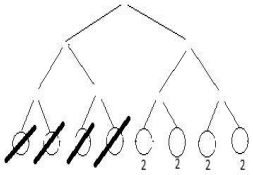


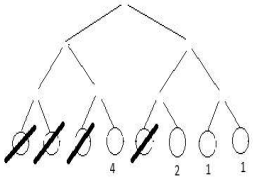
Etapa 4.5

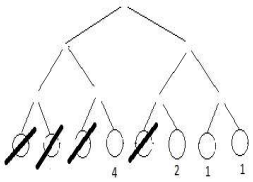


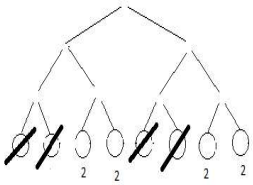
Etapa 4.6

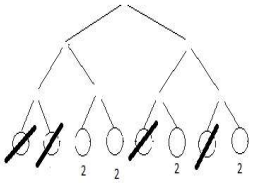


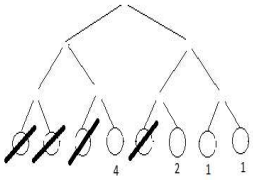
Etapa 5.1 

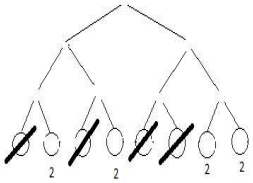
Etapa 5.2 

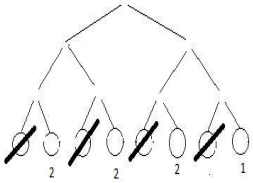
Etapa 5.3 

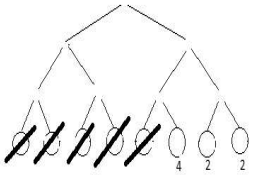
Etapa 5.4 

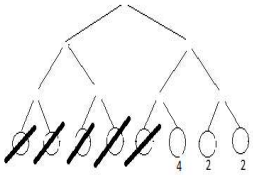
Etapa 5.5 

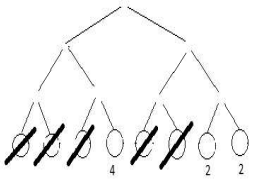
Etapa 5.6 

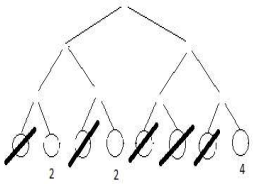
Etapa 5.7 

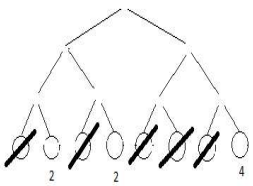
Etapa 5.8 

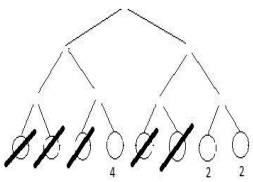
Etapa 6.1 

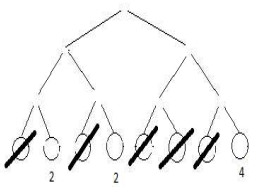
Etapa 6.2 

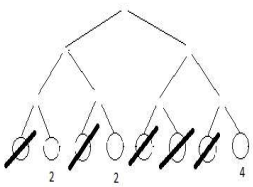
Etapa 6.3 

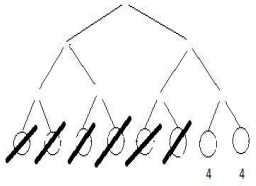
Etapa 6.4 

Etapa 6.5 

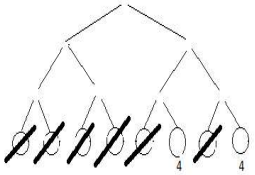
Etapa 6.6 

Etapa 6.7 

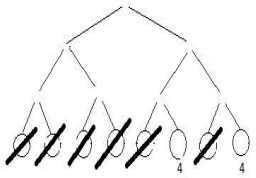
Etapa 6.8 



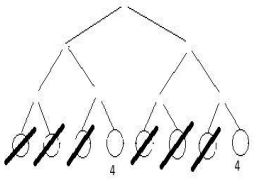
Etapa 7.1



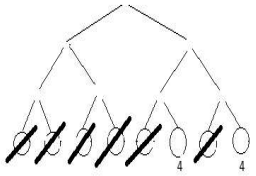
Etapa 7.2



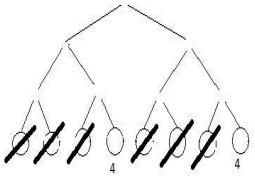
Etapa 7.3



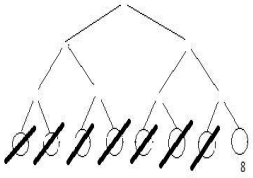
Etapa 7.4



Etapa 7.5



Etapa 7.6



Etapa 8

Una vez planteadas las posibles rupturas vamos a describirlo de forma cuantitativa, las primeras etapas serán triviales y tales que

$$\Gamma(1) = 8 \rightarrow \qquad \delta(1) = \frac{1}{8}$$

$$\Gamma(2) = 10 \rightarrow \qquad \delta(2) = \frac{1}{10}$$

A partir de esa etapa ya no es simétrica la ruptura de un elemento respecto de cualquiera de la fibra. Podremos observar que de la segunda etapa podemos tener 3 configuraciones diferentes en función de cuál sea el elemento roto. Sin embargo, podemos hacer su análisis y ver

$$\Gamma(3,1) = 12 \rightarrow \qquad P(3,1) = \frac{4}{10} = \frac{2}{5}$$
$$\Gamma(3,2) = 12 \rightarrow \qquad P(3,2) = \frac{2}{10} = \frac{1}{5}$$
$$\Gamma(3,3) = 12 \rightarrow \qquad P(3,3) = \frac{4}{10} = \frac{2}{5}$$

Aunque en esta etapa en concreto podemos comprobar que, no nos sirve como ejemplo, ya que ocurre lo mismo que en la tercera etapa del caso anterior, todas las configuraciones tienen la misma anchura, por lo que la anchura de la etapa, pesadas las probabilidades, será igual

$$\Gamma(3) = \Gamma(3,1)P(3,1) + \Gamma(3,2)P(3,2) + \Gamma(3,3)P(3,3) = 12\frac{2}{5} + 12\frac{1}{5} + 12\frac{2}{5} = 12 \tag{28}$$

$$\delta(3) = \frac{1}{12} \tag{29}$$

Ahora bien, para la cuarta y quinta etapa, en las siguientes configuraciones veremos ya el por qué de la necesidad del algoritmo en cuestión. A continuación, en esta cuarta etapa, se verán las 2 configuraciones posibles de la ruptura de la configuración 3.1, que serán las figuras 4.1 y 4.2, de la ruptura de la configuración 3.2 tendremos las figuras 4.3 y 4.4, análogamente de la ruptura de 3.3 veremos las configuraciones 4.5 y 4.6, aunque algunas serán iguales entre sí y podremos agruparlas

$$4{,}1 = 4{,}3 = A \tag{30}$$

$$P(A) = P(4{,}1)P(3{,}1) + P(4{,}3)P(3{,}2) = \frac{2}{3}\frac{2}{5} + \frac{2}{3}\frac{1}{5} = \frac{2}{5} \tag{31}$$

$$\Gamma(A) = 20 \tag{32}$$

$$4{,}2 = 4{,}5 = B \tag{33}$$

$$P(B) = P(4{,}2)P(3{,}1) + P(4{,}5)P(3{,}3) = \frac{1}{3}\frac{2}{5} + \frac{2}{3}\frac{2}{5} = \frac{2}{5} \tag{34}$$

$$\Gamma(B) = 14 \tag{35}$$

$$4{,}4 = 4{,}6 = C \tag{36}$$

$$P(C) = P(4{,}4)P(3{,}2) + P(4{,}6)P(3{,}3) = \frac{1}{3}\frac{1}{5} + \frac{1}{3}\frac{2}{5} = \frac{1}{5} \tag{37}$$

$$\Gamma(A) = 14 \tag{38}$$

Entonces tenemos ahora dos posibles diferentes anchuras, (ya que las anchuras de dos de esas configuraciones, B y C, son iguales), por ello tendremos esa anchura de la ruptura de esta etapa como

$$\Gamma(4) = \Gamma(A)P(A) + \Gamma(B)P(B) + \Gamma(C)P(C) = 20\frac{2}{5} + 14\frac{2}{5} + 14\frac{1}{5} = 20\frac{2}{5} + 14\frac{3}{5} \tag{39}$$

$$\delta(4) = 11/175 = 0{,}06285 \tag{40}$$

Podemos ver por ello que esta etapa podrá darnos dos valores de la anchura posibles, dados como $\Gamma = 20$ con un 0.4 de probabilidad y $\Gamma = 14$ con una probabilidad de 0.6.

Algo parecido sucederá para la siguiente etapa, empezando desde la posible ruptura del estado A, observaremos que tenemos dos configuraciones posibles 5.1 y 5.2, para B tendremos 5.3, 5.4 y 5.5, y finalmente para C tendremos 5.6, 5.7 y 5.8, por lo cual el procedimiento será análogo al caso anterior.

$$5{,}1 = E \tag{41}$$

$$P(E) = P(5{,}1)P(A) = \frac{4}{5}\frac{2}{5} = \frac{8}{25} \tag{42}$$

$$\Gamma(E) = 16 \tag{43}$$

$$5{,}2 = 5{,}3 = 5{,}6 = F \tag{44}$$

$$P(F) = P(5{,}2)P(A) + P(5{,}3)P(B) + P(5{,}6)P(C) = \frac{1}{5}\frac{2}{5} + \frac{4}{7}\frac{2}{5} + \frac{4}{7}\frac{1}{5} = \frac{74}{175} \tag{45}$$

$$\Gamma(F) = 22 \tag{46}$$

$$5{,}4 = G \tag{47}$$

$$P(G) = P(5{,}4)P(B) = \frac{2}{7}\frac{2}{5} = \frac{4}{35} \tag{48}$$

$$\Gamma(G) = 16 \tag{49}$$

$$5{,}5 = 5{,}7 = H \tag{50}$$

$$P(H) = P(5{,}5)P(B) + P(5{,}7)P(C) = \frac{1}{7}\frac{2}{5} + \frac{2}{7}\frac{1}{5} = \frac{4}{35} \tag{51}$$

$$\Gamma(H) = 16 \tag{52}$$

$$5{,}8 = I \tag{53}$$

$$P(I) = P(5{,}8)P(C) = \frac{1}{7}\frac{1}{5} = \frac{1}{35} \tag{54}$$

$$\Gamma(I) = 16 \tag{55}$$

Con lo cual veremos que, la anchura final de esta etapa vendrá dada como

$$\Gamma(5) = \Gamma(E)P(E) + \Gamma(F)P(F) + \Gamma(G)P(G) + \Gamma(H)P(H) + \Gamma(I)P(I) = \tag{56}$$

$$= 16\frac{8}{25} + 22\frac{74}{175} + 16\frac{4}{35} + 16\frac{4}{35} + 16\frac{1}{35} = 16\frac{101}{175} + 22\frac{74}{175} \tag{57}$$

Donde, como en la etapa anterior, tenemos dos posibles valores de la anchura (16 y 22) dados por esas probabilidades(101/175 y 74/175). Y un resultado tal que

$$\delta(5) = 1703/30800 = 0{,}055292207 \tag{58}$$

En el resto de etapas este suceso no ocurre como se va a comprobar, de la ruptura de E tendremos 6.1, de F tendremos las figuras 6.2, 6.3 y 6.4, de la ruptura de G podremos ver 6.5, de H tendremos 6.6 y 6.7 y de I tendremos 6.8, por lo que de nuevo agrupamos tal que

$$6{,}1 = 6{,}2 = J \tag{59}$$

$$P(J) = P(6,1)P(E) + P(6,2)P(F) = \frac{1208}{1925} \tag{60}$$

$$\Gamma(J) = 24 \tag{61}$$

$$6,3 = 6,6 = L \tag{62}$$

$$P(L) = P(6,3)P(F) + P(6,6)P(H) = \frac{258}{1925} \tag{63}$$

$$\Gamma(L) = 24 \tag{64}$$

$$6,4 = 6,5 = 6,7 = 6,8 = M \tag{65}$$

$$P(M) = P(6,4)P(F) + P(6,5)P(G) + P(6,7)P(H) + P(6,8)P(I) = \frac{459}{1925} \tag{66}$$

$$\Gamma(M) = 24 \tag{67}$$

Pero realmente las probabilidades ahora solo influyen en el camino, ya que realmente la anchura es la misma, luego siempre tendrá el mismo valor, como ya hemos apreciado en varias ocasiones previamente

$$\Gamma(6) = \Gamma(J)P(J) + \Gamma(L)P(L) + \Gamma(M)P(M) = 24\frac{1208}{1925} + 24\frac{258}{1925} + 24\frac{459}{1925} = 24 \tag{68}$$

$$\delta(6) = 1/24 = 0{,}04166... \tag{69}$$

Continuamos con la séptima etapa, de la ruptura de J tendremos 7.1 y 7.2, de la ruptura de L obtendremos como posible 7.3 y 7.4, y finalmente de M se verán 7.5 y 7.6, por lo que agrupando de nuevo

$$7,1 = 7,3 = N \tag{70}$$

$$P(N) = P(7,1)P(J) + P(7,3)P(L) \tag{71}$$

$$\Gamma(N) = 32 \tag{72}$$

$$7{,}2 = 7{,}5 = O \tag{73}$$

$$P(O) = P(7{,}2)P(L) + P(7{,}5)P(M) \tag{74}$$

$$\Gamma(O) = 32 \tag{75}$$

$$7{,}4 = 7{,}6 = P \tag{76}$$

$$P(P) = P(7{,}4)P(L) + P(7{,}6)P(M) \tag{77}$$

$$\Gamma(P) = 32 \tag{78}$$

Y entonces

$$\Gamma(7) = \Gamma(N)P(N) + \Gamma(O)P(O) + \Gamma(P)P(P) = 32 \tag{79}$$

$$\delta(7) = \frac{1}{32} = 0{,}03125 \tag{80}$$

Para la última etapa, las rupturas de N, O y P coinciden, como era de esperar, en una única configuración tal que es la configuración de la etapa 8

$$P(8) = 1 \tag{81}$$

$$\Gamma(8) = 64 \tag{82}$$

$$\delta(8) = \frac{1}{64} = 0{,}0015625 \tag{83}$$

De esta forma podremos concluir que esos dos posibles resultados para las anchuras de esas dos etapas vendrán dados con esas proporcionalidades, por lo cual queda probado que en un caso tan sencillo como es el de 8 elementos, tendremos que realizar cierta cantidad de estadística usando MonteCarlo, de tal forma que el programa, por el hecho de promediar a tantas rupturas posibles, nos acabe dando esas probabilidades en forma de la cantidad correcta de anchuras en esas dos etapas.

Anexo 2. Código

```
#include <stdio.h >
#include <stdlib.h>
#include <math.h>
#include <time.h>
#include <string.h>
#defne NormRANu (2.3283063671E-10F)
#defne verFallo
extern double Random(void);
extern void ini_ran(int SEMILLA);
unsigned int irr[256];
unsigned char ind_ran,ig1,ig2,ig3; unsigned int ir1;
double Random(void){
double r;
ig1=ind_ran-24;
ig2=ind_ran-55;
ig3=ind_ran-61;
irr[ind_ran]=irr[ig1]+irr[ig2];
ir1=(irr[ind_ran]^irr[ig3]);
ind_ran++;
r=ir1*NormRANu;
return r;
}
void ini_ran(int SEMILLA){ int INI,FACTOR,SUM,i;
srand(SEMILLA);
INI=SEMILLA;
FACTOR=67397;
SUM=7364893;
```

```
for(i=0; i<256; i++){
INI=(INI*FACTOR+SUM);
irr[i]=INI;
}
ind_ran=ig1=ig2=ig3=0;
}
void inicializaArbolyVector(int matrix[p+1][N],double vectorCargas[N],int sigma0){
int i,j;
for(i=0;i<p+1;i++){
for(j=0;j<N;j++){
matrix[i][j]=1;
vectorCargas[j]=sigma0;//Se repite p+1 veces la inicializacion, pero al final
}
}
/* for(j=0;j<N;j++){
matrix[0][j]=j;
} */ } //Con 1=sano, 0=roto, y veriamos asi inicializo un arbol sano
void muestraArbolyVector(int matrix[p+1][N],double vectorCargas[N])
int i,j;
for(i=0;i<p+1;i++){
for(j=0;j<N;j++){
//printf("
printf("n");//Se ha llegado al nal de la la de la matriz
printf("VectorCargas: ");
for(i=0;i<N;i++)
printf("
printf();
void muestraVector(double * vector) int i; for(i=0;i<N;i++) printf("vector[ printf();
void secuenciaAleatoria(int * seCal){ int aux, j,aleat;
for(j=0;j<N;j++){ seCal[j]=j;
}
```

```
//Se ha inicializado el seCal como si fuesen de 0 a N para ir indicando una sec aleatoria
for(j=0;j<N;j++){
aleat=(int)((N-1-j)*Random());
//printf("N-1-j=aux=seCal[N-1-j];
seCal[N-1-j]=seCal[aleat];
seCal[aleat]=aux;
//muestraVector(seCal);
//getchar();
}
for(j=0;j<N;j++){
//printf("seCal[
}
//printf();
}
int potenciaManual(int ci, int pi){
int res,j;
res=1;
for(j=0;j<pi;j++){
res=ci*res;
}
return res;
}
void vektorReparte(double vectorCargas[N],int matrix[p+1][N],int pi,int k,double carga,int i,int controlDeControl){
int j,l,m,jumpo,sum,zum,pasos,tantas,control;
/// NECESITO MEMORIA DINÁMICA
vektor[potenciaManual(c,pi)]
FILE * uw;
uw=fopen("vectork.txt",.at");
int* vektor;
vektor=(int*)malloc(potenciaManual(c,pi)*sizeof(int));
```

```
control=0;
for(jumpo=k;jumpo<k+potenciaManual(c,pi);jumpo++){
sum=0;
for(j=0;j<c;j++){
if(matrix[0][jumpo*c+j]>0){
sum=sum+1;
}
}
vektor[jumpo-k]=sum;
//fprintf(uw,"vektor[%d]=%d ",jumpo-k,vektor[jumpo-k]);
control=control+vektor[jumpo-k];
}
//fprintf(uw,çontrol=
//fprintf(uw,);
if(control==0){
//fprintf(uw,"#0. control=0, activo ");
if(controlDeControl==1){
//fprintf(uw,çontrolDeControl=%d ",controlDeControl);
//controlDeControl=0;
vektorReparte(vectorCargas,matrix,pi+1,(int)(k/c),carga,i,controlDeControl);
controlDeControl=0;
}
}
if(control!=0 && controlDeControl==1){
//fprintf(uw,"0. control!=0, activo loop pasos ");
for(pasos=1;pasos<pi+1;pasos++){
for(l=0;l<potenciaManual(c,pi-pasos);l++){
zum=0;
for(m=0;m<c;m++)
sum=0;
for(j=0;j<potenciaManual(c,pasos-1);j++){
```

```
if( vektor[ l*potenciaManual(c,pasos) +m*potenciaManual(c,pasos-1) + j] >0 ){
sum=sum+1;
}
}
if( sum >0){
zum=zum+1;
}
}
//A multiplicar
for(m=0;m<c;m++){
for(j=0;j<potenciaManual(c,pasos-1);j++)
if( vektor[ l*potenciaManual(c,pasos) +m*potenciaManual(c,pasos1) + j] >0 ){
vektor[ l*potenciaManual(c,pasos) +m*potenciaManual(c,pasos-1) + j] = zum*vektor[ l*potenciaManual(c,p
+m*potenciaManual(c,pasos-1) + j];
}
}
}
//Fin de la l-esima interacción
}
//Fin del paso-esimo interaccion
}
}
rolDeControl==1) //fprintf(uw,"#0. control!=0, activo loop jumpo");
for(jumpo=k;jumpo<k+potenciaManual(c,pi);jumpo++){
if(vektor[jumpo-k]>0){
for(j=0;j<c;j++){
if(matrix[0][jumpo*c+j]>0){
vectorCargas[jumpo*c+j]+=carga/vektor[jumpo-k];
}
}
}
```

```
}
}
fclose(uw);
free(vektor);
//printf("bye ");
//getchar();
void redefineMatrixyVector27(double vectorCargas[N],int matrix[p+1][N],int fibraRota,int i,int
tal){
int pi,k,subo,j;
int cuentaSanos,cuentaSanotes,soloReparteUno,controlDeControl;
double carga,cargaProbby,cargaPls,cargaProbbySuma;
matrix[0][fibraRota]=0;
carga=vectorCargas[fibraRota];
vectorCargas[fibraRota]=0;
cuentaSanos=0;
soloReparteUno=1;
int pip,potpi,csub;
FILE*v;
v=fopen("pi.txt",.at");
FILE * uw;
uw=fopen("vectork.txt",.at");
for(pi=0;pi<p;pi++)
pip=(int)(fibraRota/pow(c,pi+1));
potpi=potenciaManual(c,pi);
for(j=0;j<potenciaManual(c,pi+1);j++)
if(matrix[0][pip*potenciaManual(c,pi+1)+j])
cuentaSanos++;
if(cuentaSanos==0)
matrix[pi+1][pip]=0;
if(cuentaSanos!=0) if(soloReparteUno) soloReparteUno=0;
controlDeControl=1;
```

```
k=pip*potpi;
vektorReparte(vectorCargas,matrix,pi,k,carga,i,controlDeControl);
}
}
if(soloReparteUno==0)
break;
for(j=0;j<N;j++)
cargaPls+=vectorCargas[j];
//printf(çargaPls= %f ",cargaPls);
//fprintf(v,çargaTot= %lf",cargaPls);
fclose(v);
fclose(uw);
double cargacons;
cargacons=0.0;
for(j=0;j<N;j++)
if(matrix[0][j])
cargacons+=vectorCargas[j];
void tiempos(double * vectorBigTala, int tala) FILE* vbt;
vbt=fopen("vbt.txt","wt");
FILE* timy;
timy=fopen("tiempos.txt","wt");
//printf("hi");
int i,n;
double m,S,prev_mean;
m=0;
S=0;
n=0;
double timeMed,errorTime,desvEst;
timeMed=0.0;
for(i=0;i<tala;i++)
timeMed+=vectorBigTala[i]/tala;
```

```
for(i=0;i<tala;i++)
prev_mean=m;
n=n+1;
m=m+(vectorBigTala[i]-m)/n;
S=S+(vectorBigTala[i]-m)*(vectorBigTala[i]-prev_mean);
//errorTime=(vectorBigTala[i]-timeMed)*(vectorBigTala[i]-timeMed)/tala;
fprintf(timy," %f",vectorBigTala[i]);
desvEst=sqrt(S/tala);
errorTime=desvEst/sqrt(tala);
//errorTime=sqrt(errorTime);
fclose(vbt);
fclose(timy);
void leyDePotencias(int tala, int gamma0,int sigma0,double vectorCargas[N],int matrix[p+1][N],int
ro,double * Tgamma,double * Tdelta)
int i,j,k,fibraRota,m,seCal[N];
double gammaMayus,gammaMinus[N],probRuptura[N];
double aleat,probby;
double gaMayus[N],deltA[N];
double delt[N],cargaPls;
double * vectorDeltas4;
double * vectorDeltas5;
double * vectorBigTala;
vectorBigTala=(double*)malloc(tala*sizeof(double));
vectorDeltas4=(double*)malloc(tala*sizeof(double));
vectorDeltas5=(double*)malloc(tala*sizeof(double));
i=1;
for(j=0;j<N;j++) gaMayus[j]=0.0;
deltA[j]=0.0;
FILE*u; u=fopen("pi.txt","wt"); fclose(u); FILE*v; v=fopen("pi.txt","wt");
for(k=0;k<tala;k++)
//vectorBigTala[k]=1/(N*N);
```

```
if(k%1000==0) printf("LLEVO %d ",k);
//printf("tala= %d ",k);
for(i=1;i<N+1;i++)
//printf(.etapa %d-esima",i);
for(j=0;j<N;j++)
gammaMinus[j]=gamma0*pow( (vectorCargas[j]/sigma0) , ro);
gammaMayus=0.0;
for(j=0;j<N;j++)
if(matrix[0][j])
gammaMayus+=gammaMinus[j];
gaMayus[i]+=gammaMayus/tala;
deltA[i]+=(1.0/gammaMayus)/tala;
//vectorBigTala[k]+=1/gammaMayus;
if(i==4)
vectorDeltas4[k]=1/gammaMayus;
if(i==5)
vectorDeltas5[k]=1/gammaMayus;
//fprintf(deltas,"delta[ %d]= %f ",i,deltA[i]);
for(j=0;j<N;j++)
if(matrix[0][j])
probRuptura[j]=gammaMinus[j]/gammaMayus;
else
probRuptura[j]=0.0;
probby=0.0;
aleat=Random();
secuenciaAleatoria(seCal);
for(j=0;j<N;j++)
///montecarlo
if(matrix[0][seCal[j]])
probby+=probRuptura[seCal[j]];
if(probby>aleat)
```

```
fibraRota=seCal[j];
break;
redefineMatrixyVector27(vectorCargas,matrix,fibraRota,i,k);
cargaPls=0.0;
for(j=0;j<N;j++)
cargaPls+=vectorCargas[j];
inicializaArbolyVector(matrix,vectorCargas,sigma0); //getchar();
///TOCA TRABAJAR CON vectorBigTala
tiempos(vectorBigTala,tala);
delta45(vectorDeltas4,vectorDeltas5,tala);
free(vectorBigTala);
free(vectorDeltas4);
free(vectorDeltas5);
fclose(v);
//fclose(vh);
gaMayus[N]=0.0; for(j=0;j<N;j++)
gaMayus[N]+=vectorCargas[j];
gaMayus[N]=gaMayus[N]*gaMayus[N];
deltA[N]=1.0/gaMayus[N];
*Tgamma=0.0;
*Tdelta=0.0;
for(j=1;j<N+1;j++)
delt[j]=1.0/gaMayus[j];
*Tgamma+=delt[j];
*Tdelta+=deltA[j];
void rupturaVariante(int tala,double * Tgamma,double * Tdelta)
int matrix[p+1][N],i;
int sigma0,gamma0,ro;
double T,delta[N],vectorCargas[N];
double Tgam,Tdel;
sigma0=1;
```

```
gamma0=1;
ro=2;
inicializaArbolyVector(matrix,vectorCargas,sigma0);///Inicializar el árbol para los paráme-
tros dados //muestraArbolyVector(matrix,vectorCargas);
*Tgamma=Tgam; *Tdelta=Tdel;
int main()
c=2;//2
p=16;//20
N=potenciaManual(c,p);
double Tgamma,Tdelta;
int i,pi,pmax,tala;
tala=100;
pmax=1;
pi=p;
FILE*fallingTime;
fallingTime=fopen("weAreGoingDown.txt",.at");
FILE*deltas;
deltas=fopen("deltas.txt",.at");
for(i=0;i<pmax;i++)
p=pi+i;
N=potenciaManual(c,p); /// Defino las potencias que quiero como enteros gracias a este
//getchar();
rupturaVariante(tala,&Tgamma,&Tdelta);
///El algoritmo de ruptura se desarrolla entero en
este ///algoritmo junto con otro que llama "leyDePotencias(...) 2
este llama a redefineMatrixyVector27(...)"
//getchar();
fclose(fallingTime);
fclose(deltas);
```

(84)