PROYECTO FADA

Cristhian Botero Rodriguez

Julián David Leal Pedraza

Juan Camilo Obando Rendon

1860054-3743

1860143-3743

1859971-3743

Fundamentos de análisis y diseño de algoritmos

2020-2

Estrategias para actualización tecnológica

Para el desarrollo de las estrategias para la actualización tecnológica a un plazo de tres años vamos a utilizar los siguientes datos:

Antigüedad (años)	Costo de Mtto	Valor de venta	Venta compra
1	180	600	300
2	210	400	500
3	240	300	600
4	270	225	675

COMPRA DE MAQUINA NUEVA: 900M

A partir de estos datos elaboramos dos soluciones con dos estrategias diferentes, una estrategia voraz y una estrategia dinámica, las cuales las explicaremos a continuación.

➢ Estrategia Voraz:

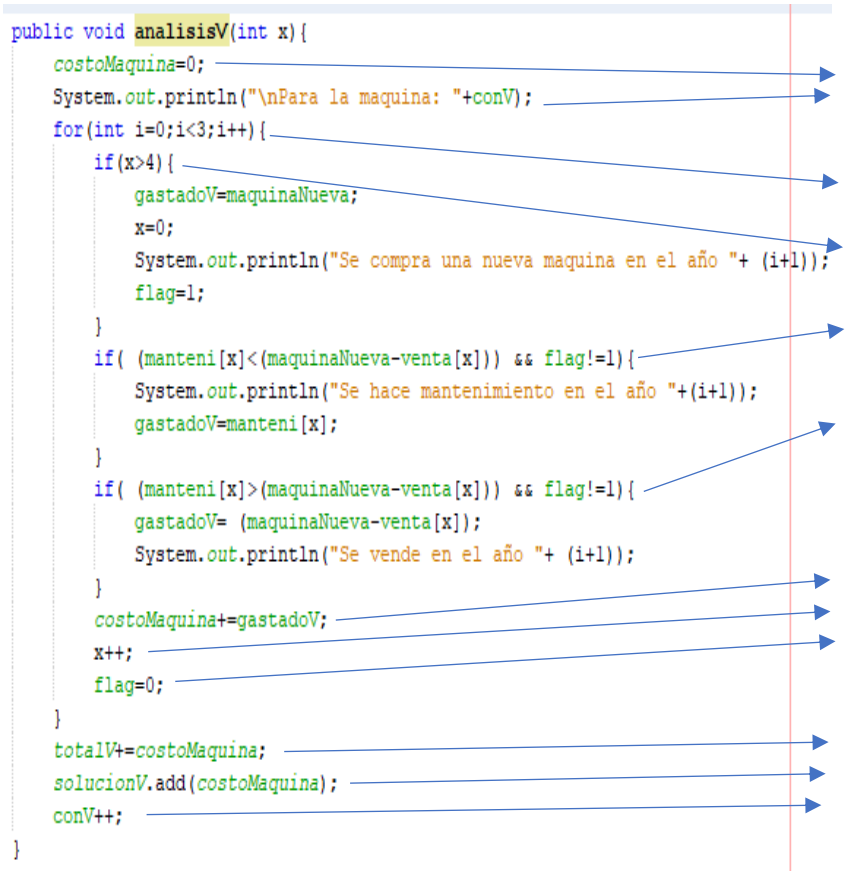
Para la estrategia voraz planteamos la siguiente solución:
Para cualquier máquina que nos pasen vamos a comparar los datos de costo de mantenimiento y el costo para comprar una maquina nueva dependiendo de la antigüedad de la máquina, nuestro objetivo es obtener el menor costo posible de esa máquina en un periodo de tres años, entonces para cada año vamos a escoger un candidato el cual tiene que ser el menor de los dos y al final para tener el obtener el costo en estos tres años vamos a sumar los tres candidatos que escogimos, cabe aclarar que si la maquina llega a tener una antigüedad de 5 años toca comprar automáticamente una maquina totalmente nueva por el costo de 900M, y reseteamos la antigüedad a 1, vamos a observar esto mediante un ejemplo:

Si recibimos una maquina con antigüedad de 2 años, tenemos que empezar a comparar los datos de ese primer año, por lo tanto, comparamos 210 con 500, el menor de estos es 210 por lo tanto se le haría mantenimiento el primer año, entonces el primer candidato seria C1=210, en el segundo año la maquina tendrá una antigüedad de tres años, por lo tanto, comparamos

240 con 600, el menor de estos es 240, por lo tanto en el segundo año le haríamos nuevamente mantenimiento, entonces este sería el segundo candidato C2=240 y para el tercer año la maquinaria tendrá una antigüedad de 4 años, comparamos 270 con 675, escogemos como candidato el 270 y le haremos mantenimiento, por lo cual el tercer candidato es C3=270.

Ahora sumaremos nuestros candidatos y este nos dará el menor costo para está maquina durante tres años, entonces seria C1+C2+C3=210+240+270= 720M el menor costo posible, en conclusión, a está maquina con antigüedad de dos años la mejor opción sería hacerle mantenimiento en el periodo de tres años.

- Complejidad computacional de la estrategia voraz:



```
public void analisisV(int x){
    costoMaquina=0;
    System.out.println("\nPara la maquina: "+conV);
    for(int i=0;i<3;i++){
        if(x>4){
            gastadoV=maquinaNueva;
            x=0;
            System.out.println("Se compra una nueva maquina en el año "+ (i+1));
            flag=1;
        }
        if( (manteni[x]<(maquinaNueva-venta[x])) && flag!=1){
            System.out.println("Se hace mantenimiento en el año "+(i+1));
            gastadoV=manteni[x];
        }
        if( (manteni[x]>(maquinaNueva-venta[x])) && flag!=1){
            gastadoV= (maquinaNueva-venta[x]);
            System.out.println("Se vende en el año "+ (i+1));
        }
        costoMaquina+=gastadoV;
        x++;
        flag=0;
    }
    totalV+=costoMaquina;
    solucionV.add(costoMaquina);
    conV++;
}
```

Costo	No instrucciones
C1	1
C2	4
C3	3
C4	3
C5	1

T(n)= Θ(1)

```
public ArrayList voraz(ArrayList m){
    for(int i=0;i<m.size();i++){
        this.analisisV(((Integer)m.get(i)));
    }
    return solucionV;
}
```

Costo	No. instrucciones
C1	n+1
C2	n
C3	1

$T(n)= C1(n+1)+C2n+C3$

$T(n)= C1n+C1+C2n+C3$

$T(n)=(C1+C2)n+(C1+C3)$

$T(n)= \Theta(n)$

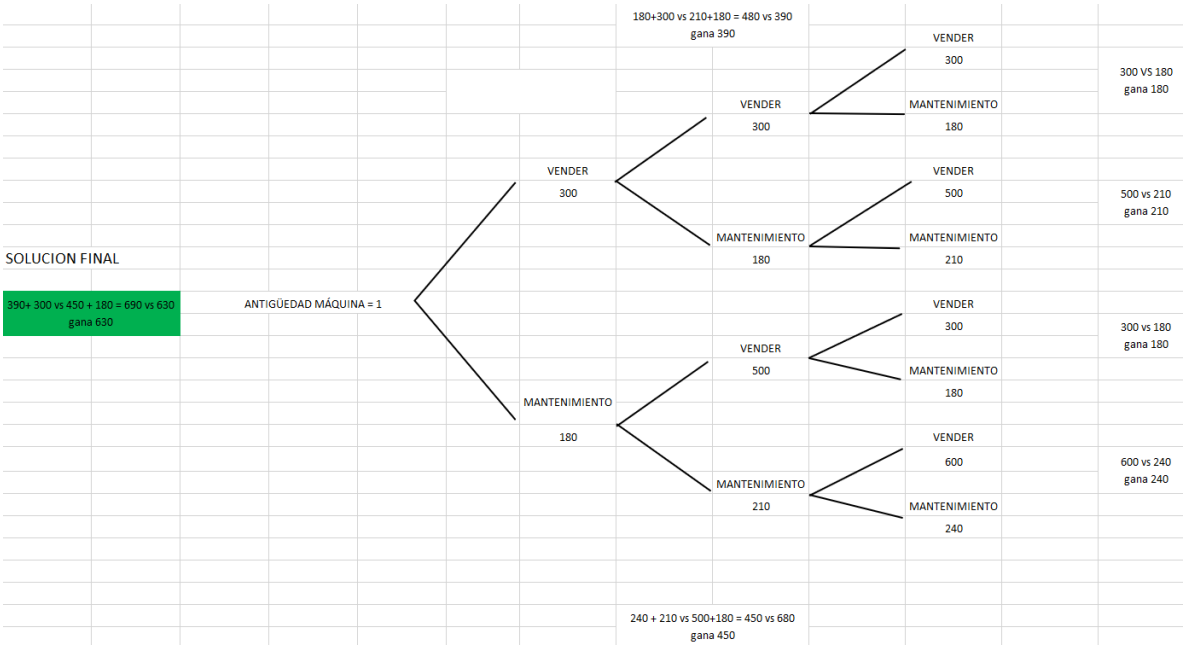
El costo total del algoritmo es $T(n)= \Theta n$, cuando la solución es para un periodo de tres años.

➢ Estrategia Dinámica y dinámica extendida:

Para la estrategia dinámica planteamos la siguiente solución:

Tenemos un algoritmo recursivo en el cual dividimos el problema en dos subproblemas más pequeños, el cual tenemos que dividir un total de tres veces que es el equivalente a los años de seguimiento que le vamos a hacer a las máquinas, cuando llega a la recursión tres, comparamos si es más rentable vender o hacerle mantenimiento a la máquina, nos quedamos con el menor costo, eso lo sumamos al subproblema de la recursión anterior y comparamos con lo obtenido en el otro subproblema, escogiendo el menor nuevamente, por última vez sumamos lo escogido con la recursión anterior que vamos a comparar con el otro subproblema de está recursión escogiendo el menor el cual va a ser la solución y nos dará el menor costo posible para una máquina.

Vamos a observar esto mediante un ejemplo:



- **Complejidad computacional de la estrategia dinámica y dinámica extendida:**

```
public int mini(int a,int b){
    if(a<b){
        return a;
    }else{
        return b;
    }
}
```

Costo	No. instrucciones
C1	1
C2	1

T(n)= C1+C2

T(n)= Θ(1)

```
public ArrayList dinamica(ArrayList m){
    for(int i=0;i<m.size();i++){
        entrada=(Integer)m.get(i);
        analisisD(entrada,0);
        totalD+=(mini(auxA,auxB));
        solucionD.add(mini(auxA,auxB));
    }
    return solucionD;
}
```

Costo	No. instrucciones
C1	n+1
C2	n
C3	1

T(n)= C1(n+1)+C2n+C3

T(n)= C1n+C1+C2n+C3

T(n)=(C1+C2)n+(C1+C3)

T(n)= Θ(n)

```
public int analisisD(int x,int a){
    if(x>4){
        gastadoA=manteni[x]+analisisD(1,a+1);
        gastadoB=venta[x]+analisisD(1,a+1);
        return mini(gastadoA,gastadoB);
    }
    if(a<=1 && x<=4){
        gastadoA=manteni[x]+ analisisD(x+1,a+1);
        if(x==this.entrada && a==0){
            auxA=gastadoA;
        }
        gastadoB=venta[x]+ analisisD(1,a+1);
        if(x==this.entrada && a==0){
            auxB=gastadoB;
        }
        //int manteni[] = {900,180,210,240,270};
        //int venta[] =   {900,300,500,600,675};
        return mini(gastadoA,gastadoB);
    }
    return mini(manteni[x],venta[x]);
}
```

T(n)=2T(n/2)+ Θ(n)

Por el método del maestro obtenemos:

A=2, B=2, F(n)=n

n.logn vs $n^{\log_2 2}$

n.logn vs n

entonces el costo computacional de la estrategia dinámica es T(n)= Θ(n.logn).

➢ Estrategia voraz extendida:

Para está estrategia extendida en la cual podemos utilizar cualquier dato que ingrese el usuario a los costos de mantenimiento, costos de venta, valor de maquina nueva y el periodo de años a la cual se desea hacerle seguimiento a la máquina, se plantea la misma solución de la estrategia voraz de comparar entre el valor de la maquina nueva menos el valor de la venta vs el costo de mantenimiento dependiendo el año de antigüedad de la máquina, escogiendo el valor más pequeño y sumándolo a una variable donde se almacenara la solución.

➢ Complejidad computacional de la estrategia voraz extendida:

```
public void analisisV(int x){
    costoMaquina=0;
    System.out.println("\nPara la maquina: "+conV);
    for(int i=0;i<años;i++){
        if(x>reinicio){
            gastadoV=maquinaNueva;
            x=0;
            System.out.println("Se compra una nueva maquina en el año "+ (i+1));
            flag=1;
        }
        if( (manteni.get(x)<=(maquinaNueva-venta.get(x))) && flag!=1){
            System.out.println("Se hace mantenimiento en el año "+(i+1));
            gastadoV=manteni.get(x);
        }
        if( (manteni.get(x)>(maquinaNueva-venta.get(x))) && flag!=1){
            gastadoV= (maquinaNueva-venta.get(x));
            System.out.println("Se vende en el año "+ (i+1));
        }
        costoMaquina+=gastadoV;
        x++;
        flag=0;
    }
    totalV+=costoMaquina;
    solucionV.add(costoMaquina);
    conV++;
}
```

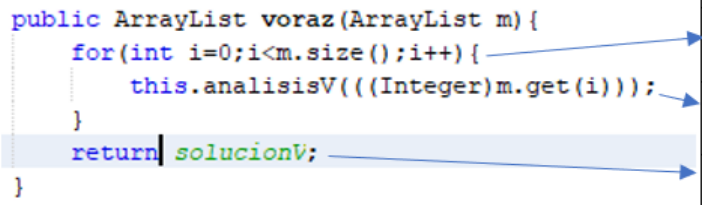
Costo	No instrucciones
C1	1
C2	m+1
C3	O(m)
C4	O(m)
C5	O(m)
C6	1

C1+C2(m+1)+C3m+C4m+C5m+C6

T(n)= C1+C2m+C2+C3m+C4m+C5m+C6

T(n)=(C2+C3+C4+C5)m+(C1+C2+C6)

T(n)= Θ(m)



Costo	No. instrucciones
C1	n+1
C2	m.n
C3	1

T(n)= C1(n+1)+C2m.n+C3

T(n)= C1n+C1+C2m.n+C3

T(n)=C1n+C2m.n+C1+C3

T(n)= Θ(n.m)

El costo total del algoritmo es T(n)= Θ(n.m), donde n es el número de máquinas y m son los años de seguimiento.