

quindi esistono A e B tali

che hanno  $P(A) > 0$  quindi

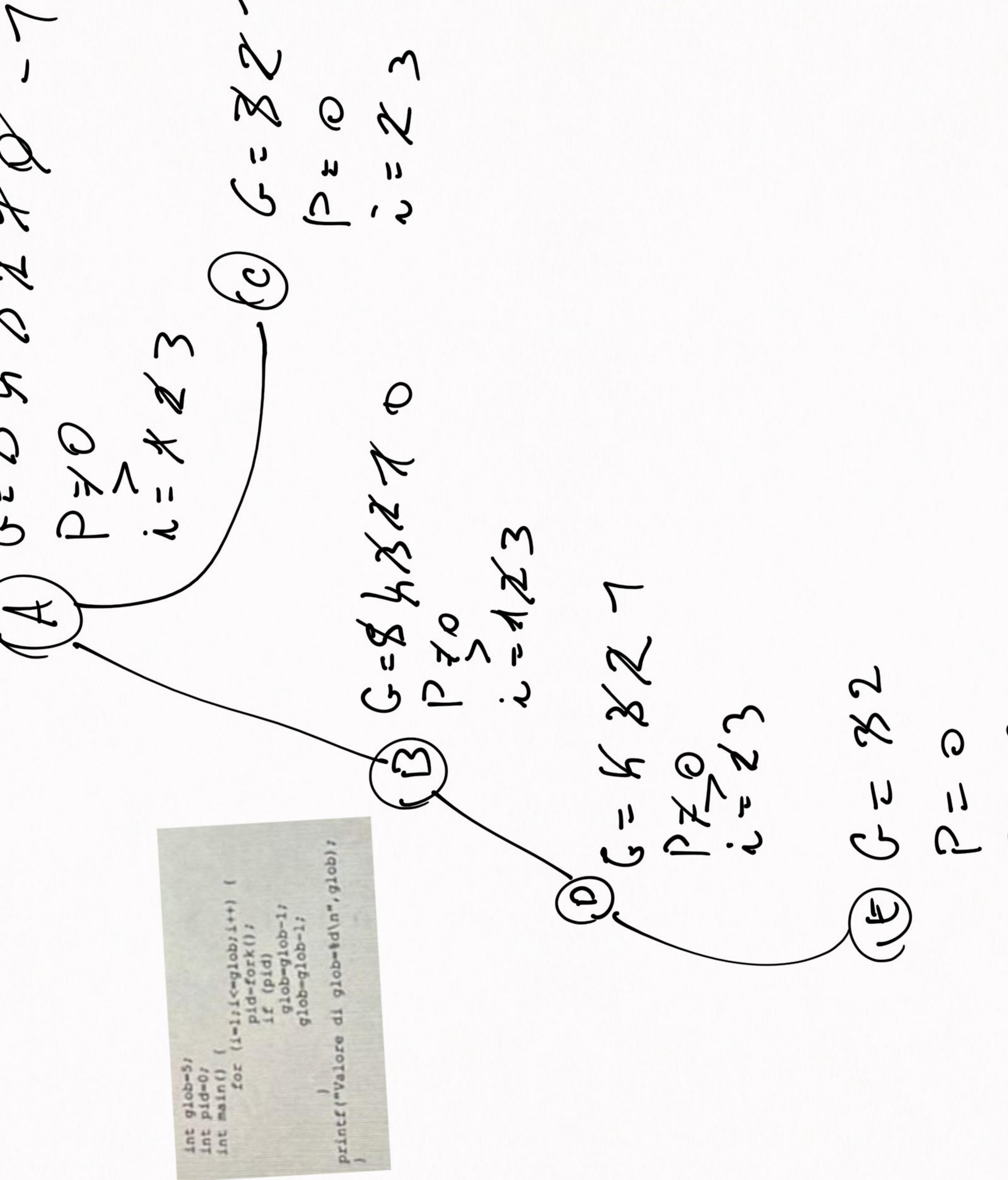
! fork

$\epsilon^* \geq$

$$(A) G = 3\sqrt{2} \quad P \geq 0$$

$$(B) G = 2\sqrt{2} \quad P = 0$$

$$(C) G = 4\sqrt{2} \quad P = 0$$



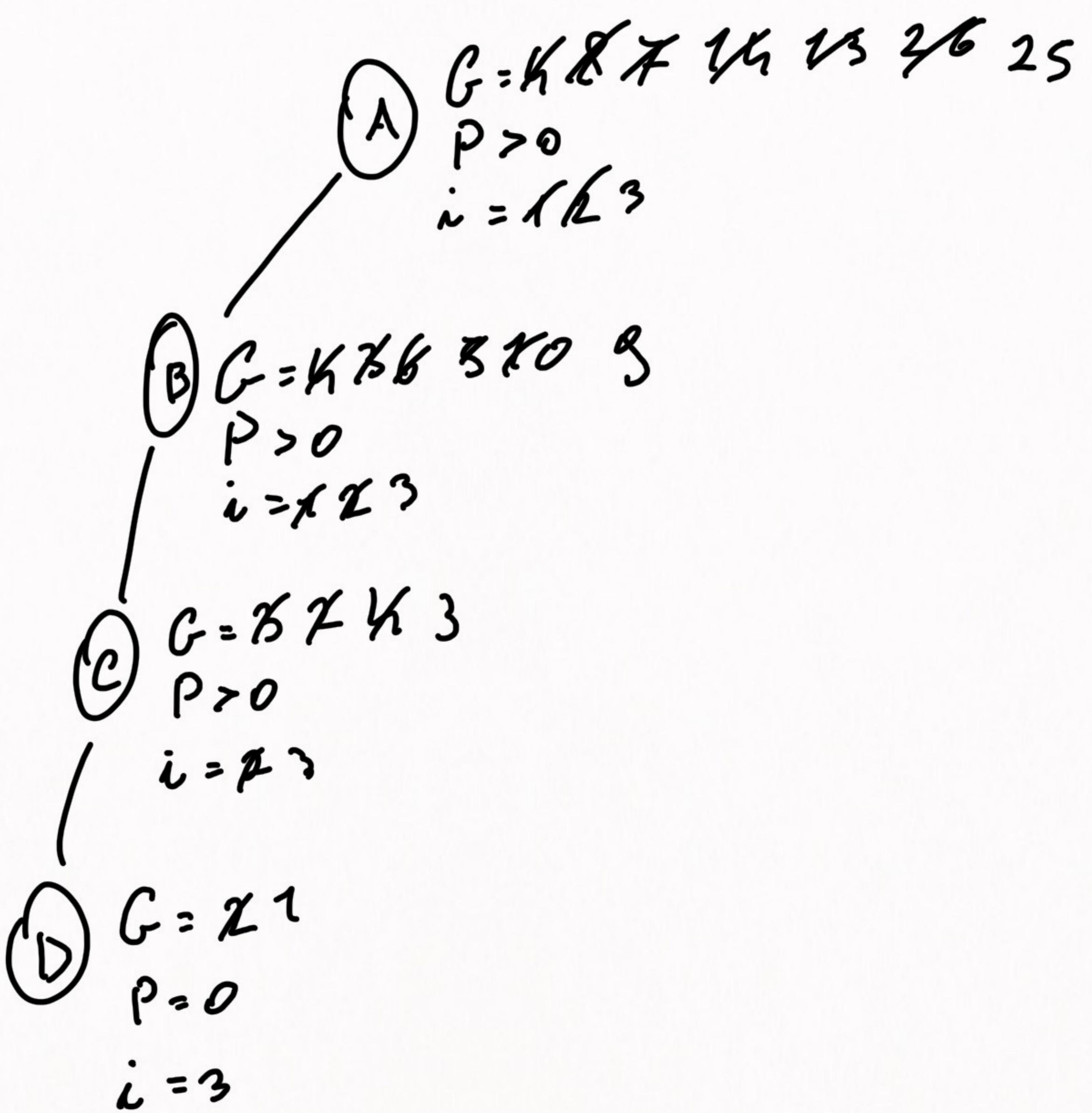
```

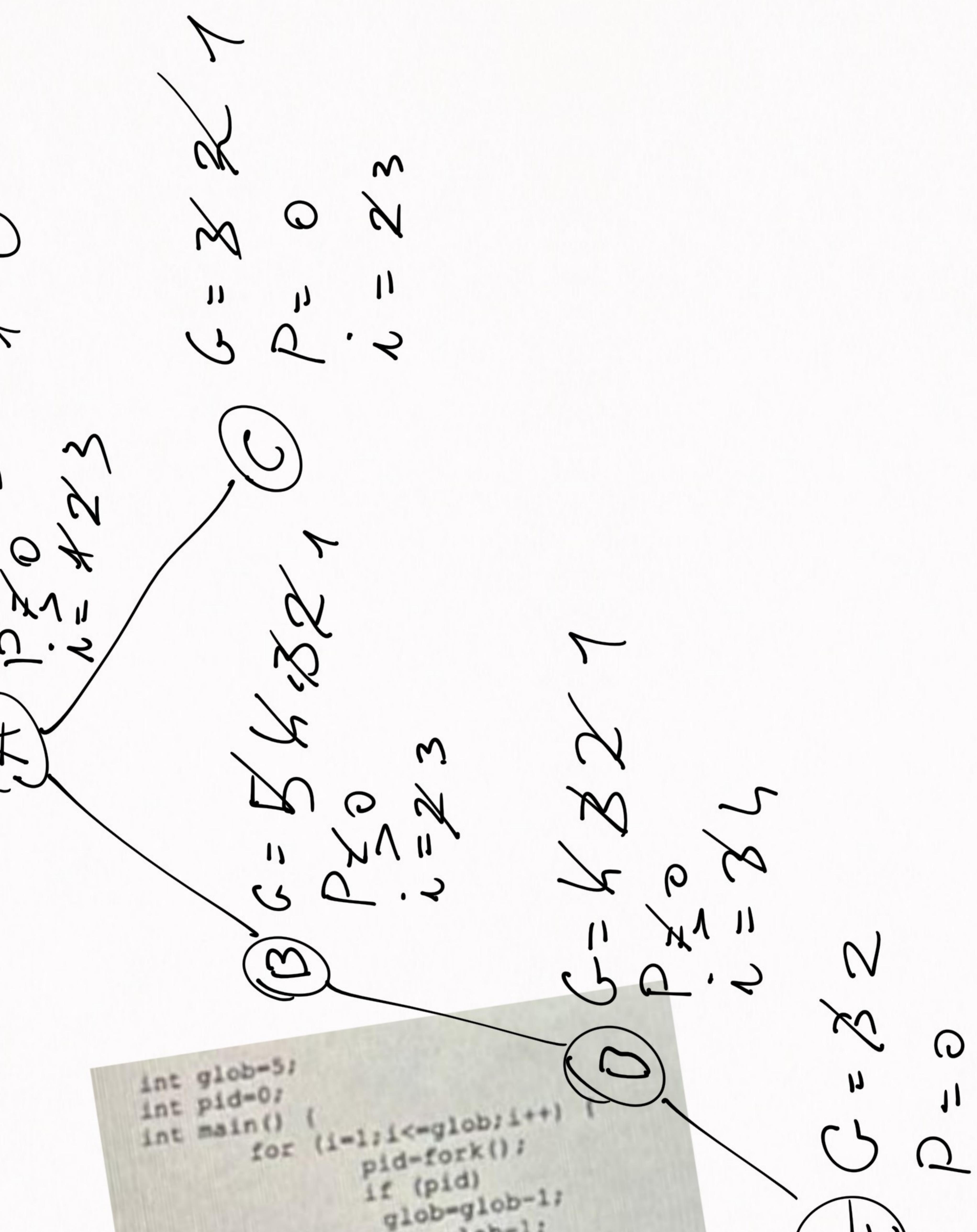
int glob = 4;
int pid = 0;

int main ()
{
    int i;
    for (i = 1; i < 4; i++)
    {
        if (pid == 0)
        {
            pid = fork();
        }

        if (pid != 0)
        {
            glob = glob * 2;
        }
        glob = glob - 1;
    }
    printf ("process %d glob = %d i = %d\n", getpid(),

```





```

int glob=5;
int pid=0;
int main() {
    for (i=1;i<=glob;i++) {
        pid=fork();
        if (pid)
            glob=glob-1;
        glob=glob-1;
    }
    printf("Valore di glob=%d\n",glob);
}
  
```

$*_z = \text{Processo Figura} \Rightarrow \text{PID} = 0$

Forza ( $i$ ) = -  $\text{PID}$   $\text{PAD}_i$   $\vec{r}_i \cdot \vec{\text{PID}}$   
- risorsa  $i$   
- Area dati: GLOBAC

```

int GLOBAC = 5;
int PID = 0;
int main () {
    int i = 0;
    for (i = 1; i < 3; i++) {
        PID = force(i);
        if (PID == 0) {
            GLOBAC = GLOBAC * 2;
            sleep(i + 1);
        }
        GLOBAC = GLOBAC * 1;
        /* i < GLOBAC si incrementa
           per tutta */
    }
}

```

1) Processo PAD<sub>0</sub>

$G = 5$   
 $PID = 0$   
 $i = 0$

2)  $G = 10$   
 $PID = 0$   
 $i = 1$

$G = 10$   
 $i = 1$



3

$$i = X^2$$

$$G = 567$$
  
 $PID \geq 0$

$$i = 2$$

$$G = 61213$$
  
 $PID = 0$

(A)

$$i = X^2$$
  
 $G = 567$   
 $PID \geq 0$

(B)

(D)

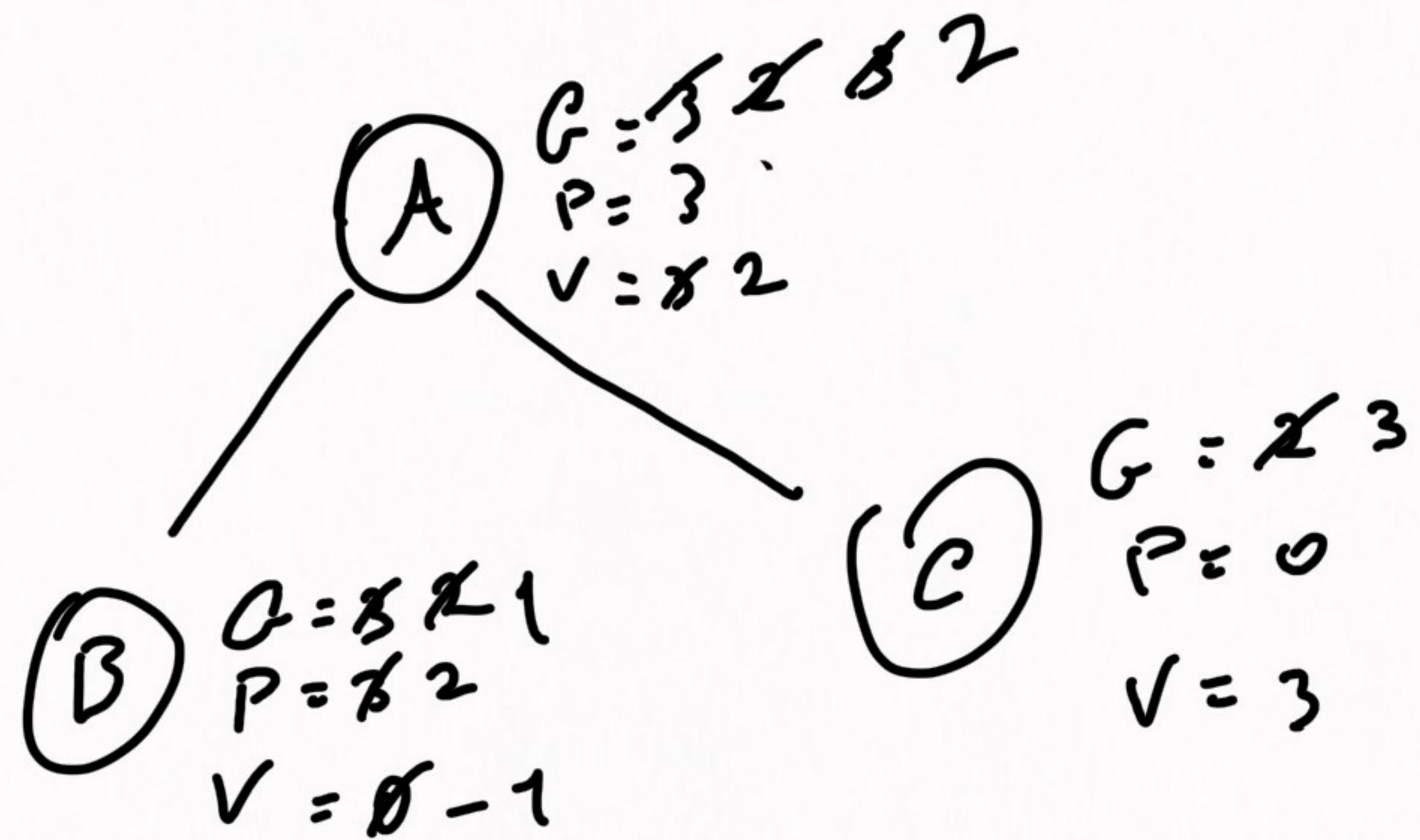
$$i = 2$$
  
 $PID = 0$

$$G = 112123$$

MARIO VISTA 012400 2266

Costruire l'albero dei processi ed indicare per ciascun processo il valore finale delle variabili glob, pid e var.

```
1. int glob=3;
2. int pid=3;
3. int var=3;
4.
5. int main(){
6.     var=fork();
7.     glob--;
8.
9.     if (var){
10.         pid=fork();
11.         glob++;
12.     }
13.     if (pid){
14.         glob--;
15.         var--;
16.     }
17.     if (!glob){
18.         var=fork();
19.         pid++;
20.     }
21. }
22. printf("Valore di glob=%d\n",glob);
23. printf("Valore di pid=%d\n",pid);
24. printf("Valore di var=%d\n",var);
25.
26.
27. }
```



`if( !( pid == fork() ) )` è equivalente

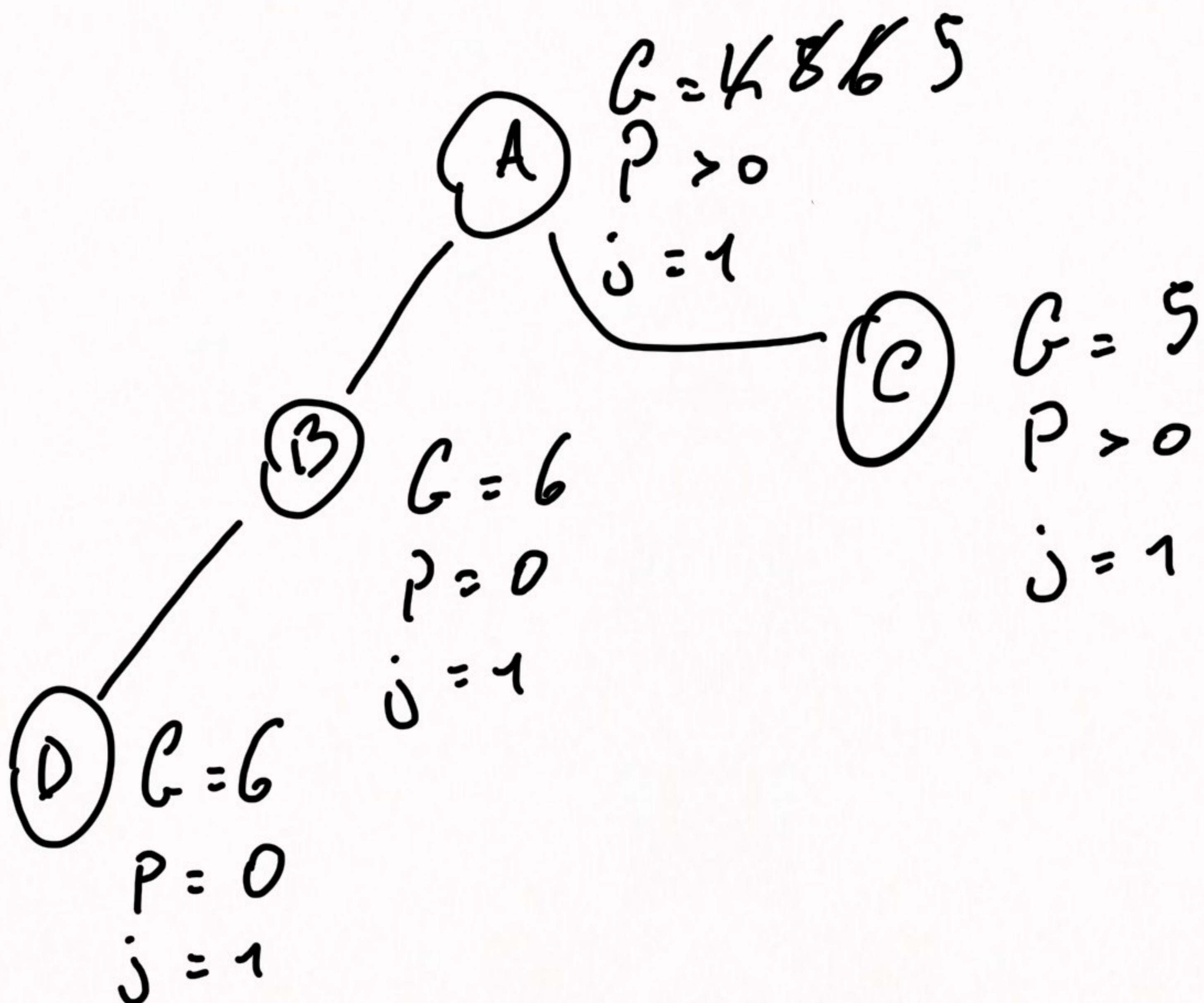
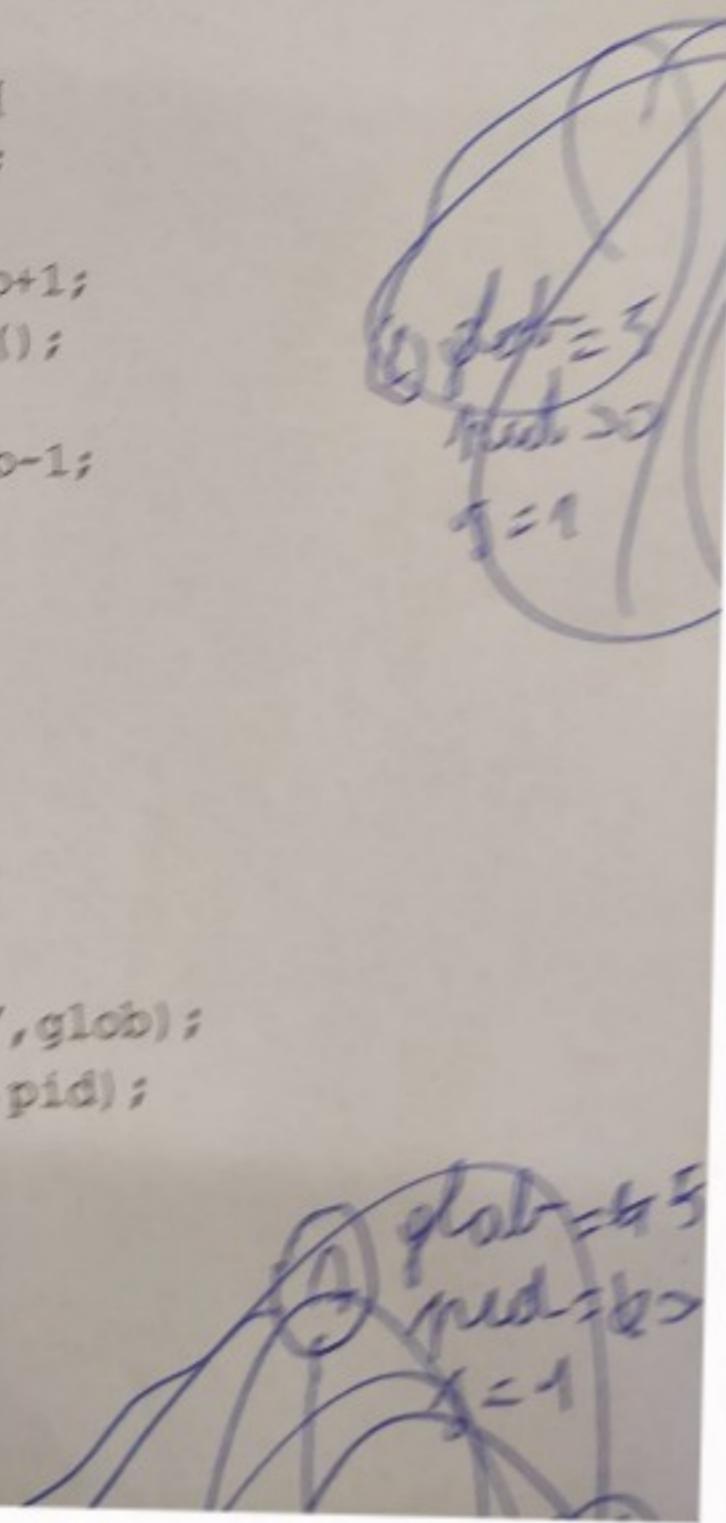
o `pid = fork()` e poi `! pid`

Ovviamente entra solo il figlio

Costruire l'albero dei processi ed indicare per ciascuna finale delle variabili glob e pid.

```
int glob=4;
int pid=0;
int main()
{
    for (j=1;j<2;j++) {
        glob=glob+j;
        if (!pid)
            glob=glob+1;
            pid=fork();
        if (pid)
            glob=glob-1;
            fork();
    }

    if (!pid)
        fork();
    if (pid)
        glob--;
    printf("glob=%d\n",glob);
    printf("pid=%d\n",pid);
}
```



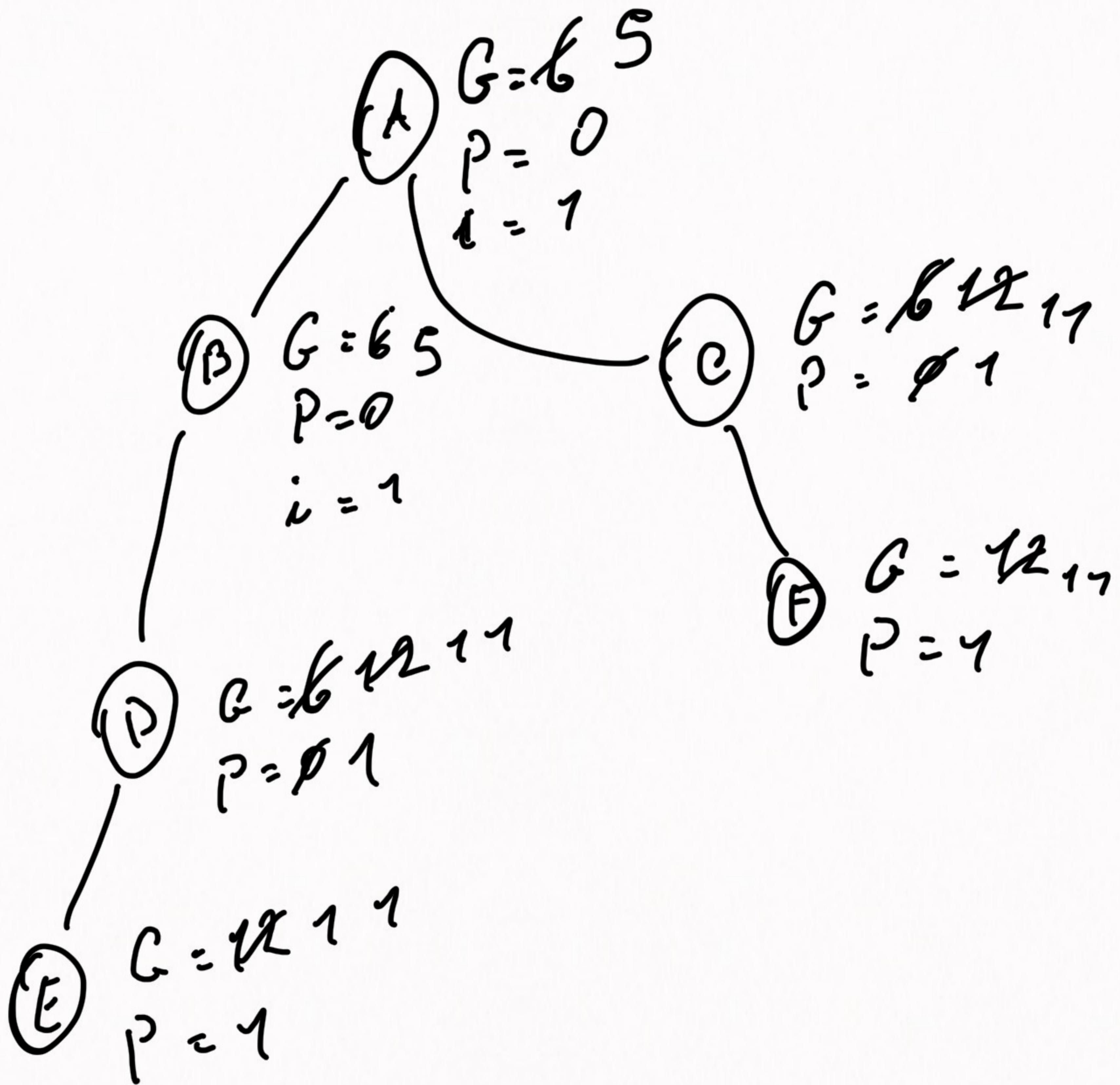
JACOMINO Giuse

Costruire l'albero dei processi ed indicare per ciascun processo il valore finale della variabile glob

```

int glob=6;
int pid=0;
fork();
for (i=1; i<2; i++) {
    if(!fork()){
        glob=glob*2;
        pid=1;
    }
    if(pid)
        fork();
    glob=glob-1;
}
printf("Valore di glob=%d\n",glob);
    
```

(A)  $\xrightarrow{g=6}$

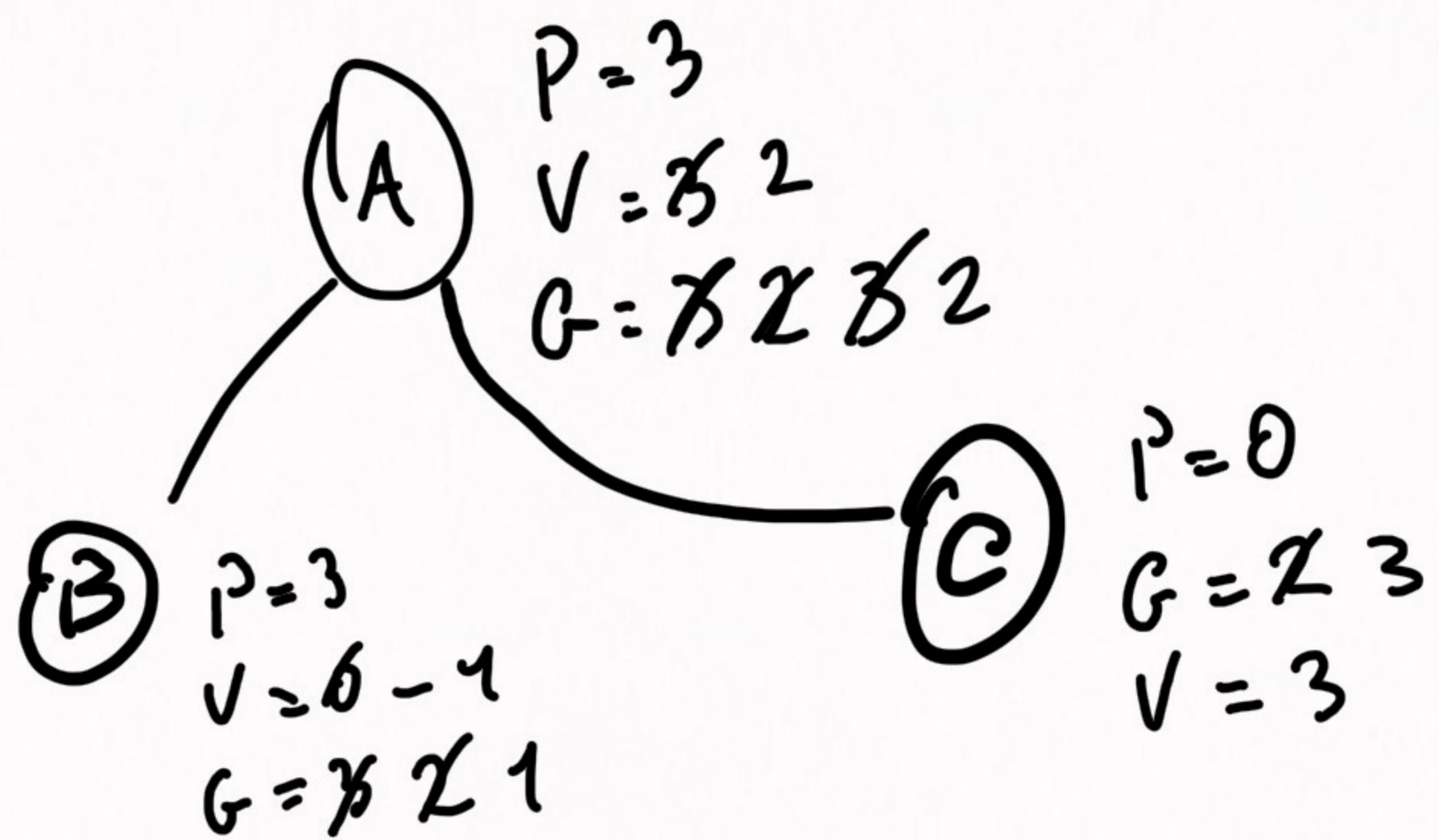


MARIO VISTIN 0124002266

Costruire l'albero dei processi ed indicare per ciascun processo il valore finale delle variabili glob, pid e var.

```

1. int glob=3;
2. int pid=3;
3. int var=3;
4.
5. int main(){
6.     var=fork();
7.     glob--;
8.
9.     if (var){
10.         pid=fork();
11.         glob++;
12.     }
13.     if (pid){
14.         glob--;
15.         var--;
16.     }
17.     if (!glob){
18.         var=fork();
19.         pid++;
20.     }
21. }
22. printf("Valore di glob=%d\n",glob);
23. printf("Valore di pid=%d\n",pid);
24. printf("Valore di var=%d\n",var);
25.
26.
27. }
```



su un bancone. I boccali sono prelevati da M cameriere può portare 2 boccali per volta, se boccali per volta. Formire una soluzione con s

### 2. 6 punti

Elencare le differenze tra un kernel che utilizza (a) un monokernel.

### 3. 9 punti (Obbligatorio)

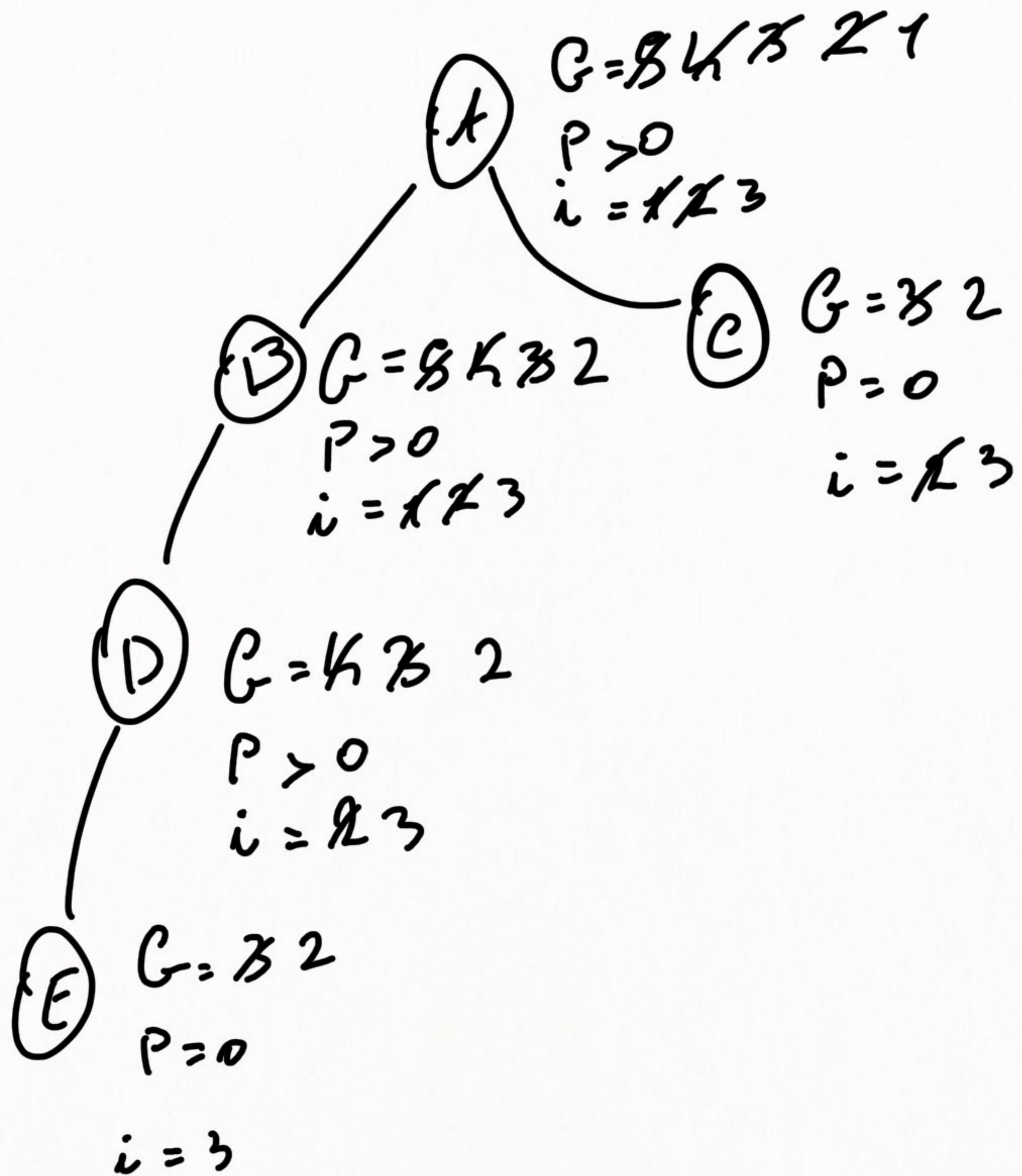
Descrivere l'albero dei processi creato con il seguente codice:

```
int glob=5;
int pid=0;
int main() {
    for (i=1;i<=glob;i++) {
        pid=fork();
        if (pid)
            glob=glob-1;
        glob=glob-1;
    }
    printf("Valore di glob=%d\n",glob);
}
```

### 4. 6 punti

Spiegare l'importanza delle informazioni gestibili tra i processi e come funziona il meccanismo di setuid/seteuid.

21°C  
Preval. nuvol.



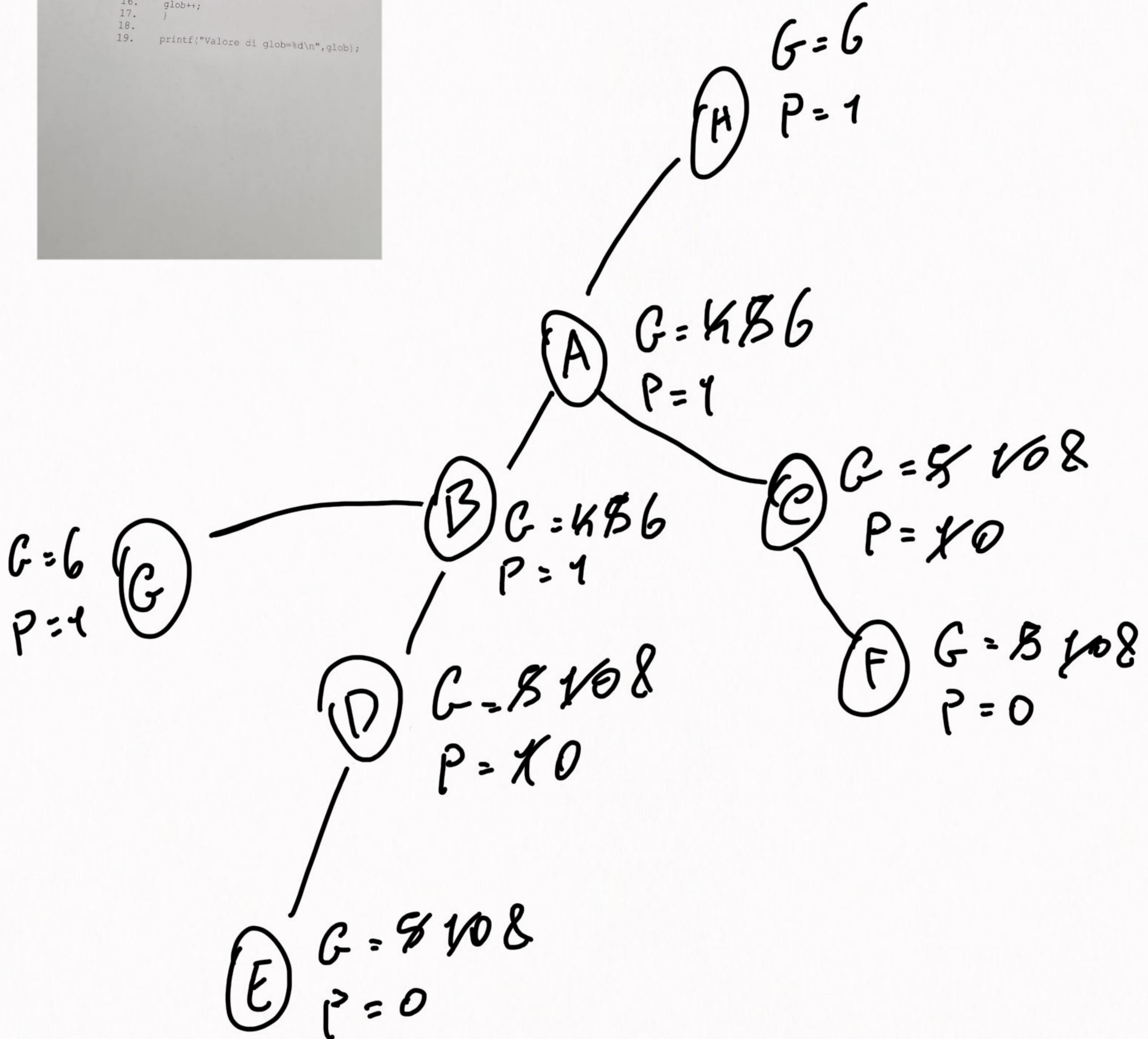
*giusto  
controllato*

Costruire l'albero dei processi ed indicare per ciascun processo il valore finale di glob.

```

1. int glob=4;
2. int pid=1;
3.
4. fork();
5. glob++;
6.
7. if(!fork()){
8.   pid=fork();
9.   glob=glob*2;
10.  pid=0;
11. }
12. if (!pid)
13.   glob=glob-2;
14. else {
15.   fork();
16.   glob++;
17. }
18.
19. printf("Valore di glob=%d\n",glob);

```



$\star_1 = \text{Processo Figlio} \Rightarrow Pid = 0$

Fork () =  
- Pid Padre: Pid  
- risorsa: i  
- Area dati: GLOBAL

```
int GLOBAL = 5;  
int Pid = 0;
```

```
int main () {
```

```
    int i = 0;
```

```
    for (i = 1; i < 3; i++)
```

```
        Pid = fork();
```

```
        if (Pid == 0)
```

```
            GLOB = GLOB * 2;  
            sleep (i + 1);
```

```
        }
```

```
    GLOB = GLOB / 2;
```

```
    } // 3 i < GLOB si incrementa per tutta
```

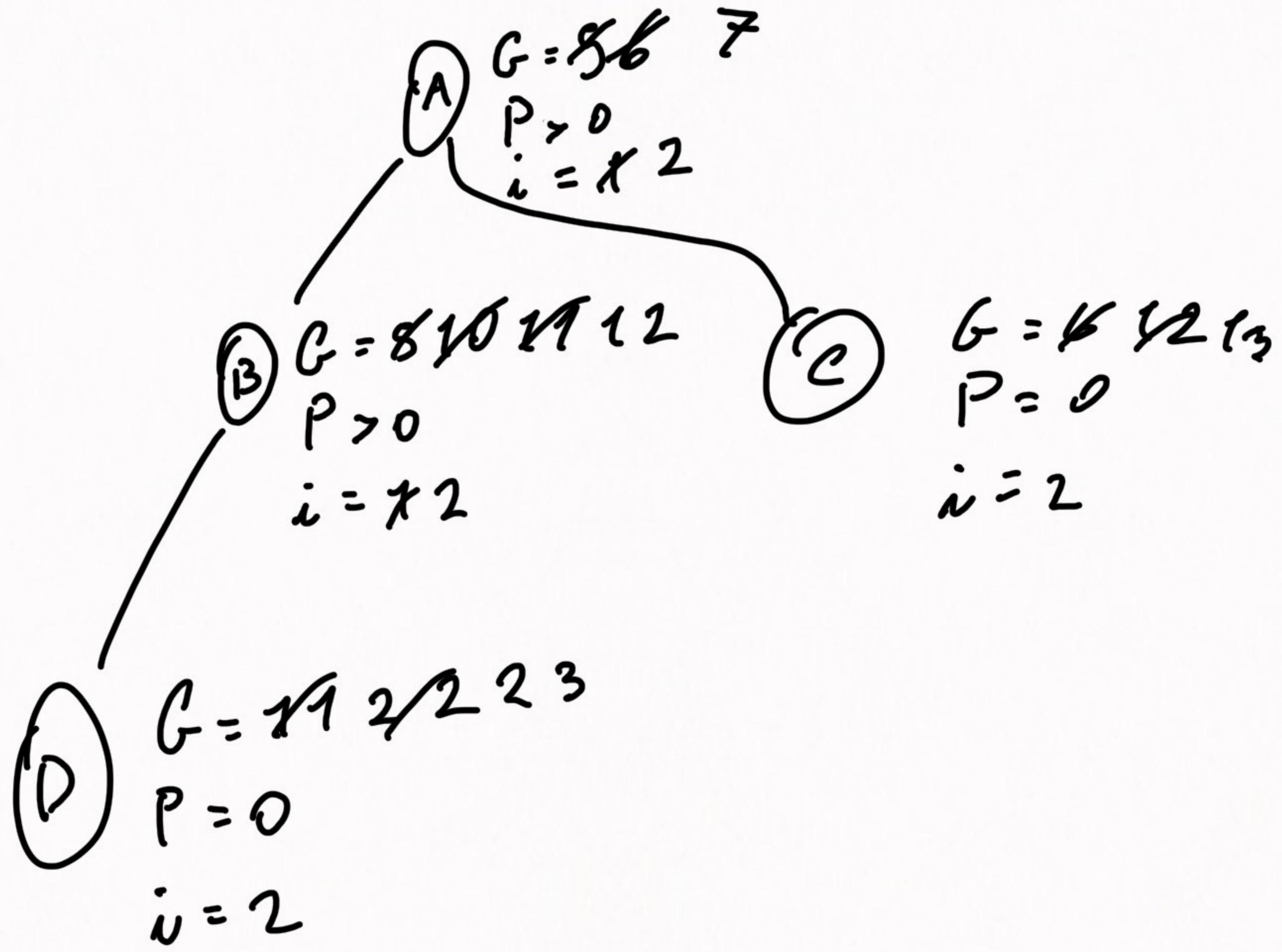
```
    }
```

1) Processo Padre

$G = 5$   
 $Pid = 0$   
 $i = 0$

2)  $G = 10$   
 $Pid > 0$   
 $i = 1$

$Pid = 0$   
 $G = 10$   
 $i = 1$

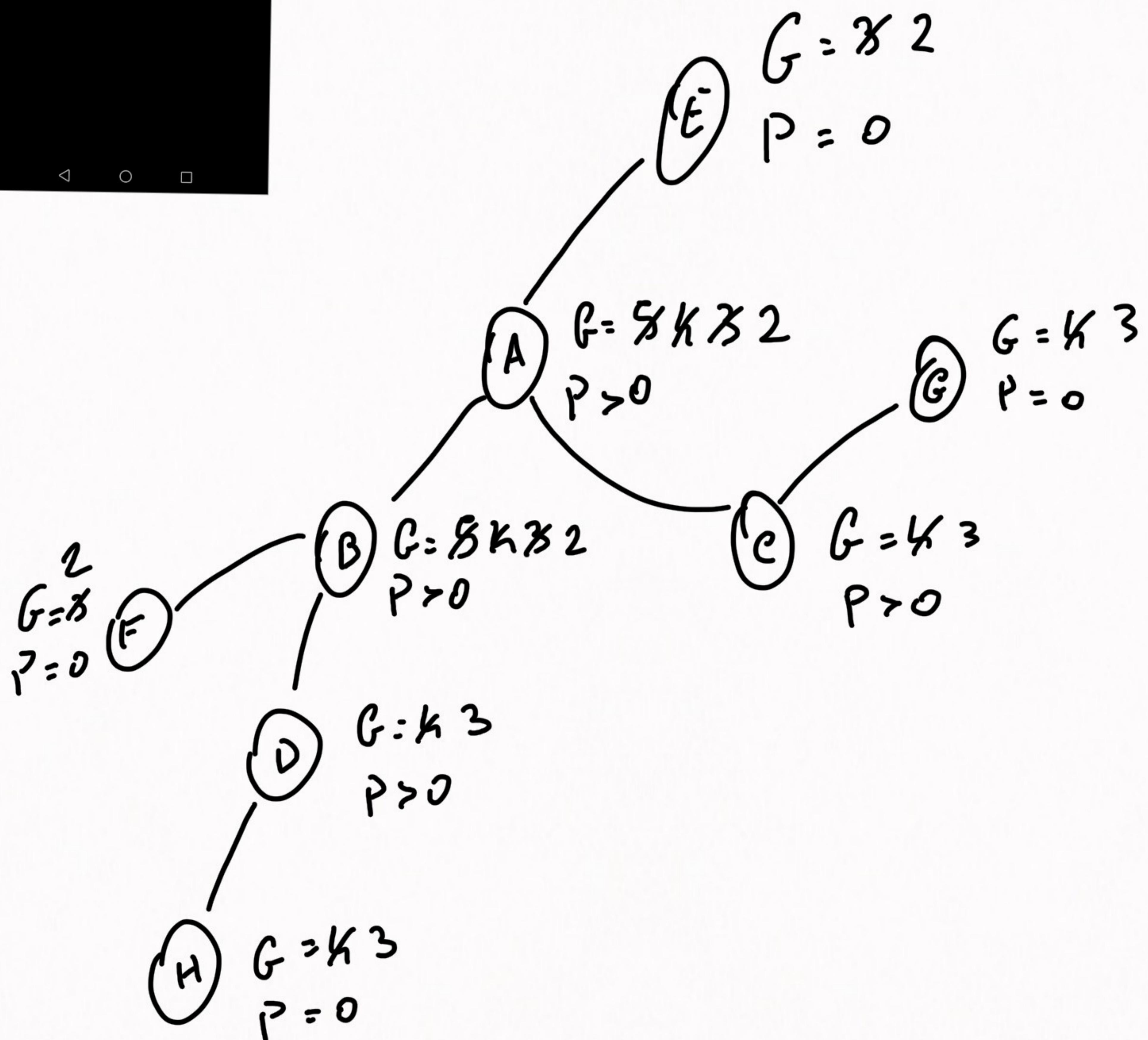


vodafone IT 79% 13:04

```
← albero31Luglio.c : 
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

int main()
{
    int glob = 5, pid = 0;
    fork();
    glob--;
    if( fork() )
        glob--;
    if( pid == 0 )
    {
        pid = fork();
        glob--;
    }
    printf("\n RIGA 24 pid = %d\tppid = %d\n", getpid(), getppid(),
glob);
}
```

*Gusto controllato*



vodafone IT 79% 13:04

```
← albero14Dicembre.c :
```

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int main()
{
    int glob = 5;
    int pid = 0;

    pid = fork();

    glob--;
    fork();

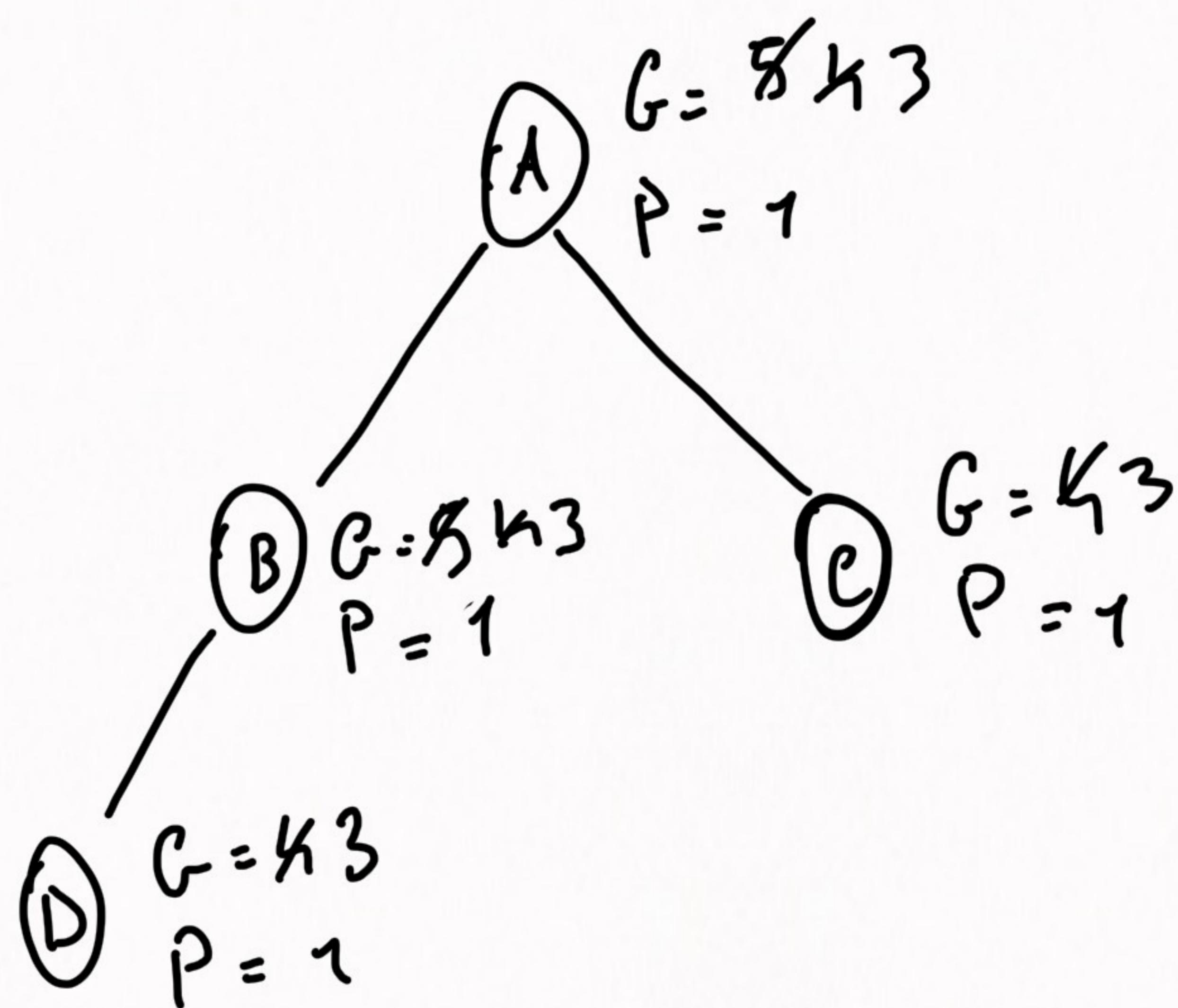
    glob--;
    pid = 1;

    if( !pid )
    {
        glob--;
        fork();
    }

    printf("\n ALLA FINE pid = %d\n Ppid = %d \n", getpid(), getppid(), glob);

}
```

*Giusto Controllato*



vodafone IT 79% 13:04

```
← albero9Novembre.c :
```

```
#include <unistd.h>
#include <stdio.h>

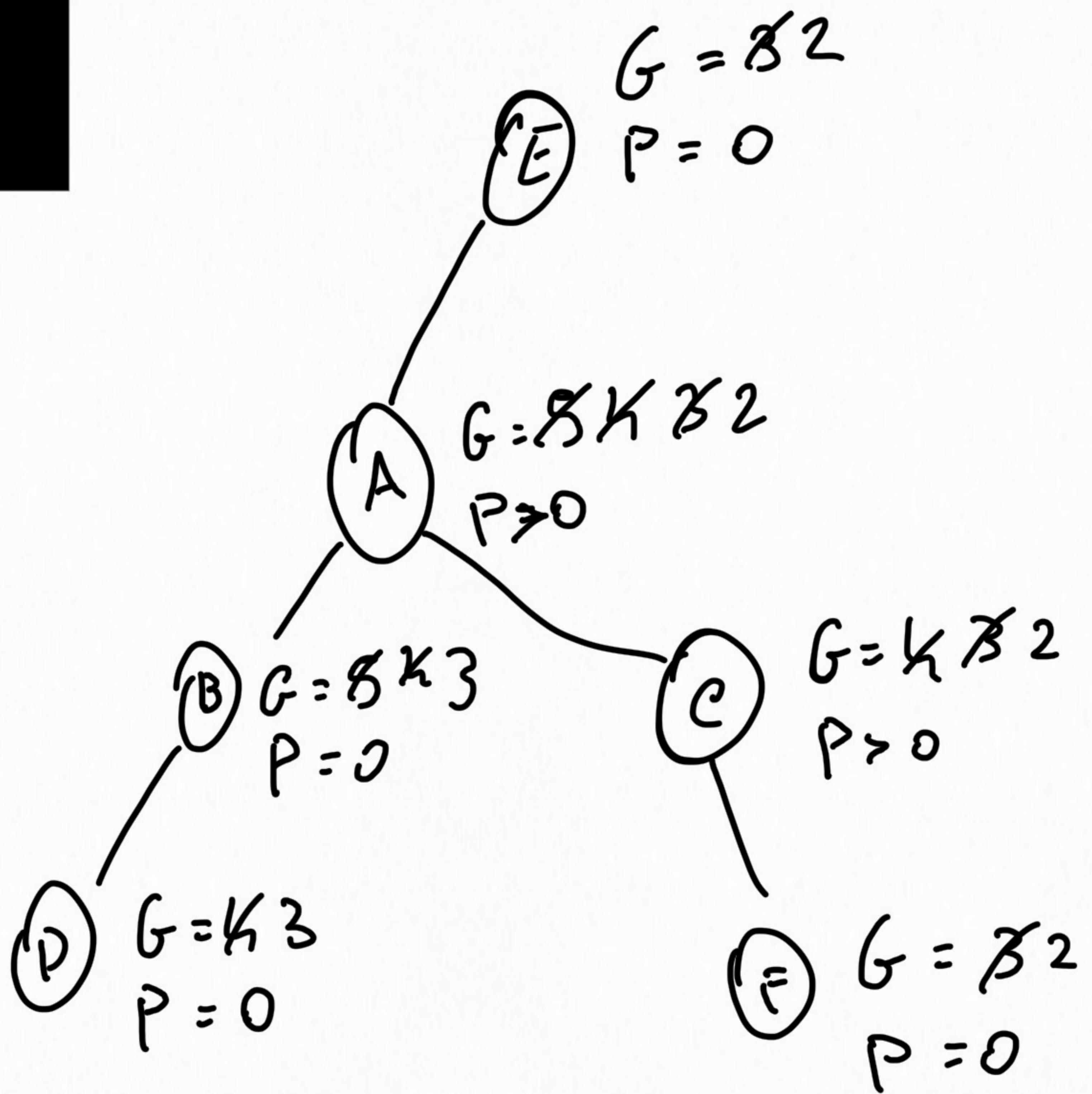
int main()
{
    int g = 5;
    int p = 0;

    p = fork();
    g--;
    fork();
    g--;

    if( p > 0 )
    {
        p = fork();
        g--;
    }

    printf("\n RIGA 23 pid = %d\tPpid = %d
\tpglob=%d \tp=%d\n", getpid(), getppid(), g,
p);
}
```

*Risultato Controllato*



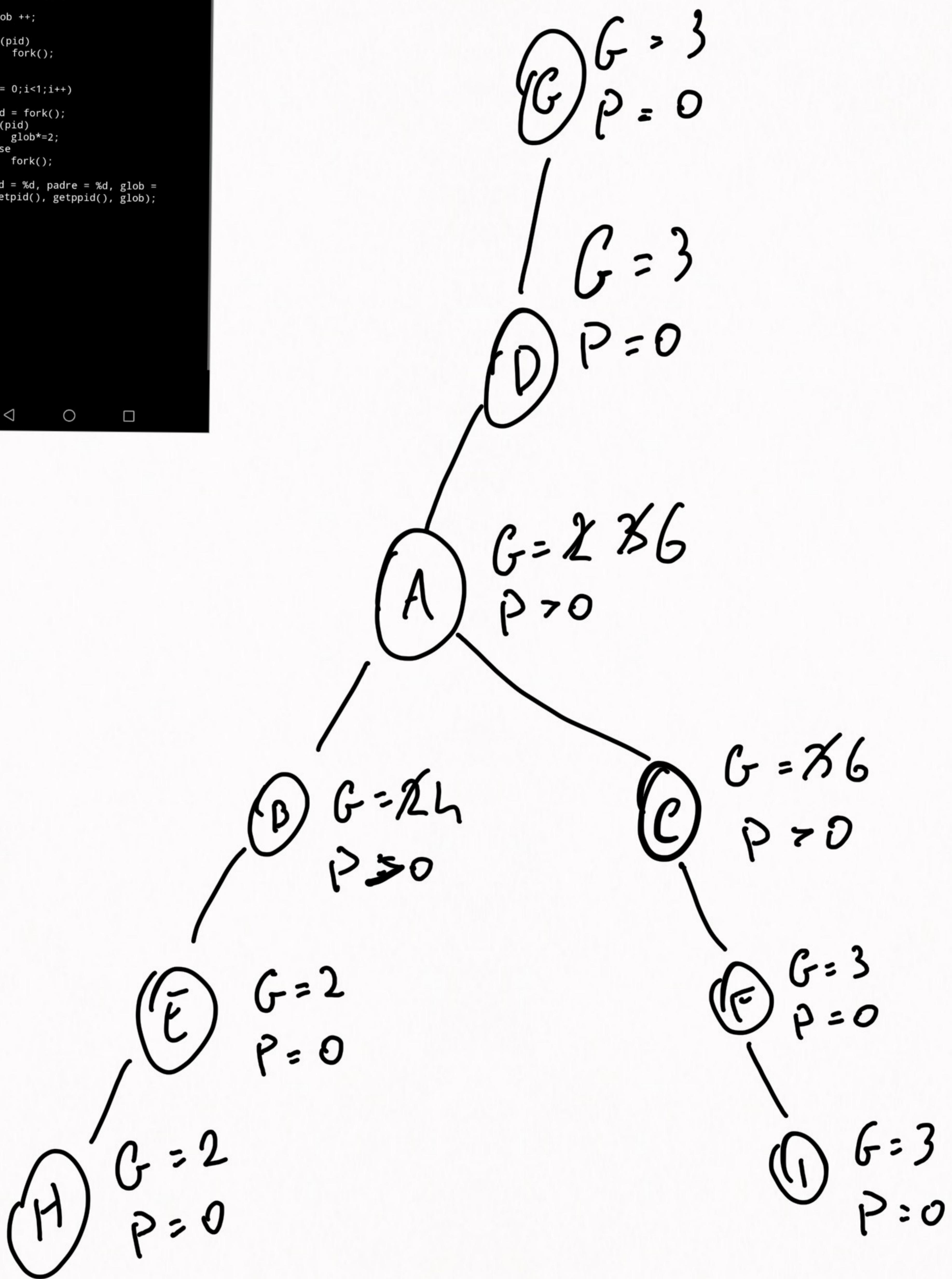
vodafone IT 79% 13:04

```
← alberino.c
```

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

int main()
{
    int glob = 2, pid = 0, i;
    if(pid = fork() )
    {
        glob++;
        if(pid)
            fork();
    }
    for(i = 0;i<1;i++)
    {
        pid = fork();
        if(pid)
            glob*=2;
        else
            fork();
    }
    printf("pid = %d, padre = %d, glob = %d\n\n", getpid(), getppid(), glob);
}
```

*Giusto Controllato*



vodafone IT 79% 13:04

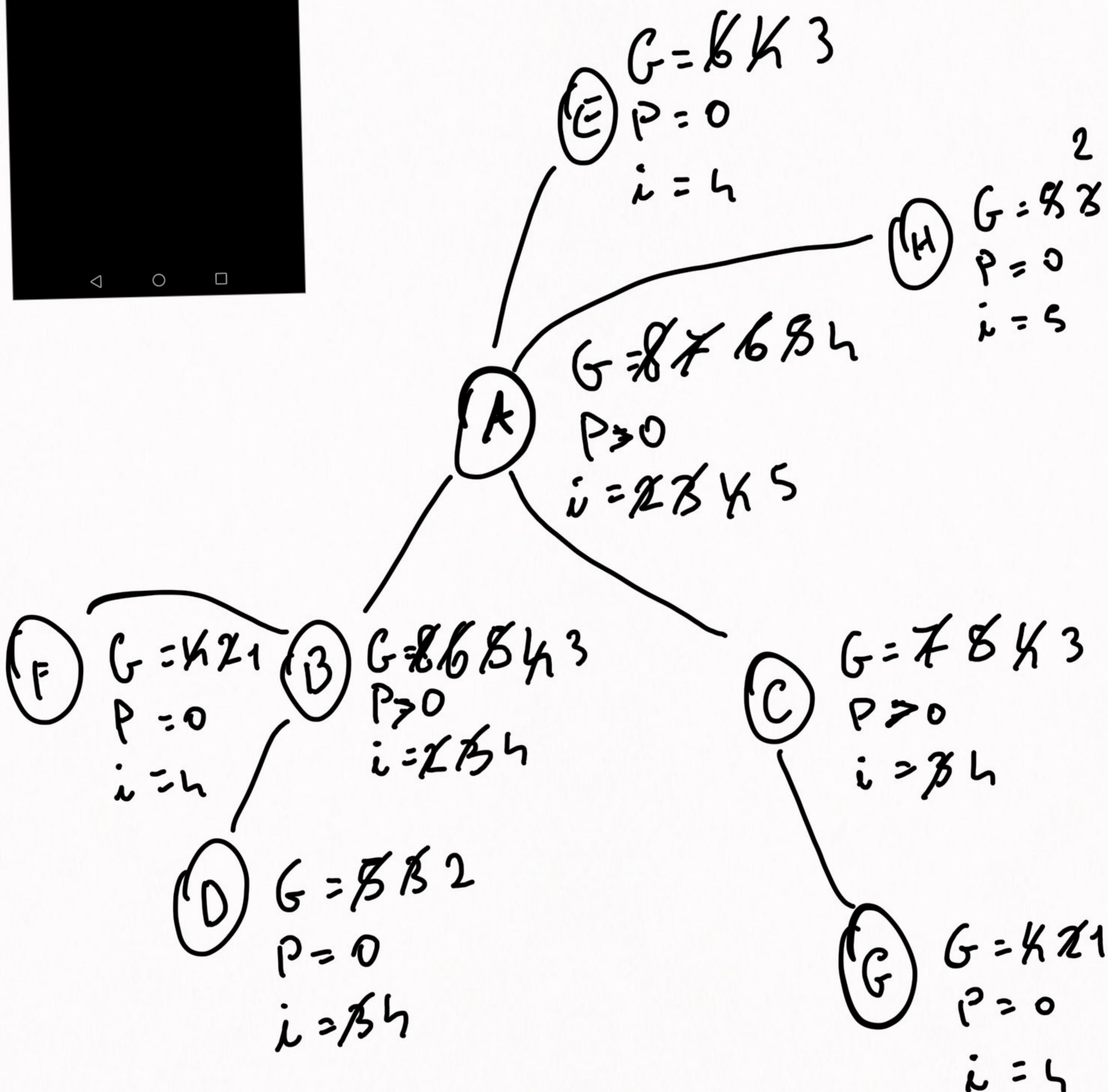
← albero1.c

```
#include <stdio.h>
#include <stdlib.h>
#include <unistd.h>

int glob=8;
int pid=0;
int i;
int main()
{
    for (i=2;i<=glob;i++)
    {
        pid = fork();
        if (pid==0)
            glob=glob-2;
        glob--;
        printf(" i vale %d Valore di
glob=%d\n",i, glob);
        printf("\n\n");
    }
}
```

◀ ○ □

# Livello Controllato



```

vodafone T 13:04
← alberello.c
X
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

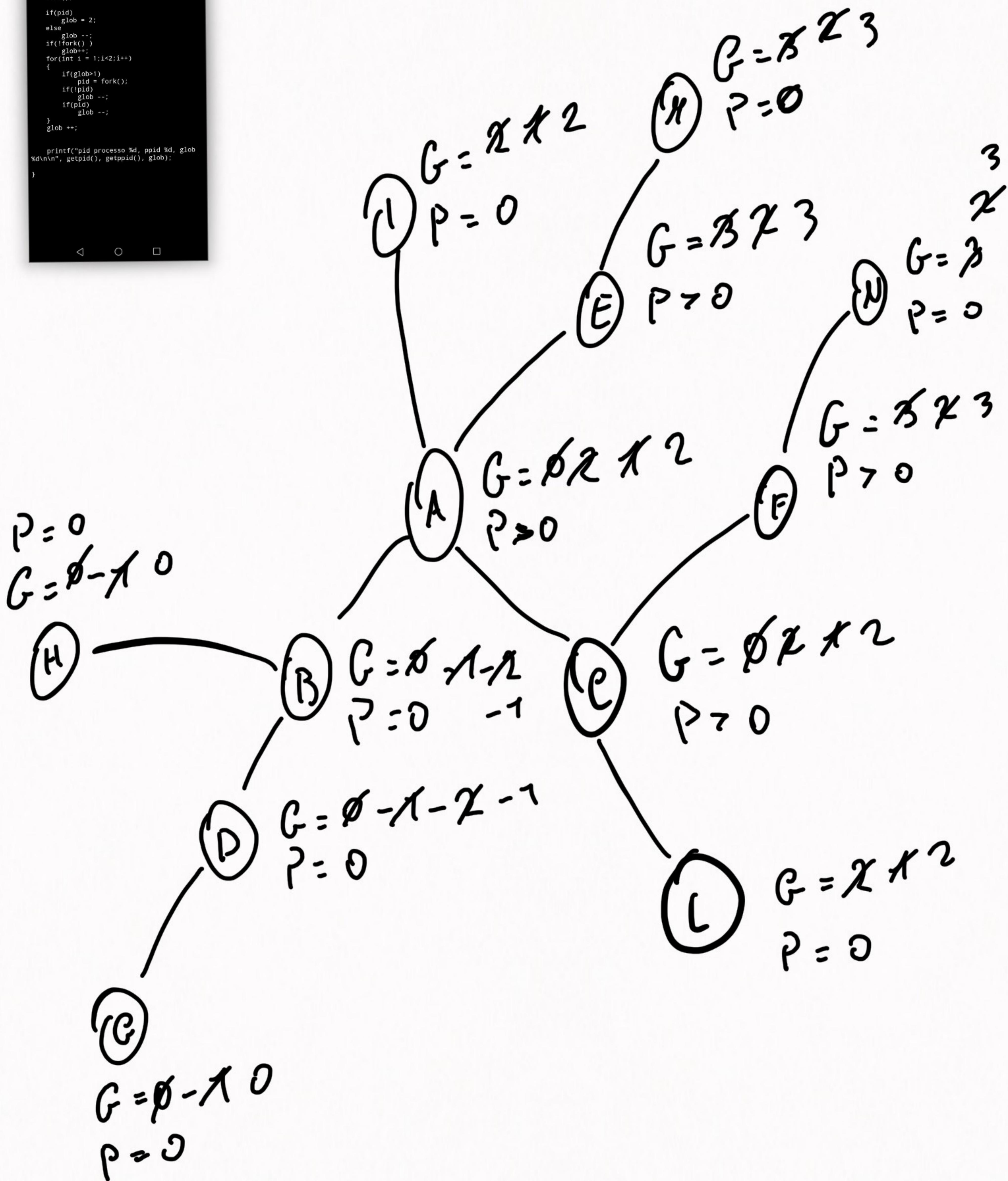
int main()
{
    int glob = 0;
    int pid = 0;
    pid = fork();
    fork();

    if(pid)
        glob = 2;
    else
        glob--;
    if(!fork())
        glob++;
    for(int i = 1;i<2;i++)
    {
        if(glob>1)
            pid = fork();
        if(!pid)
            glob--;
        if(pid)
            glob--;
    }
    glob++;
}

printf("pid processo %d, ppid %d, glob %d\n", getpid(), getppid(), glob);
}

```

# Giusto Controllato



vodafone IT 79% 13:04

← alberello.c :

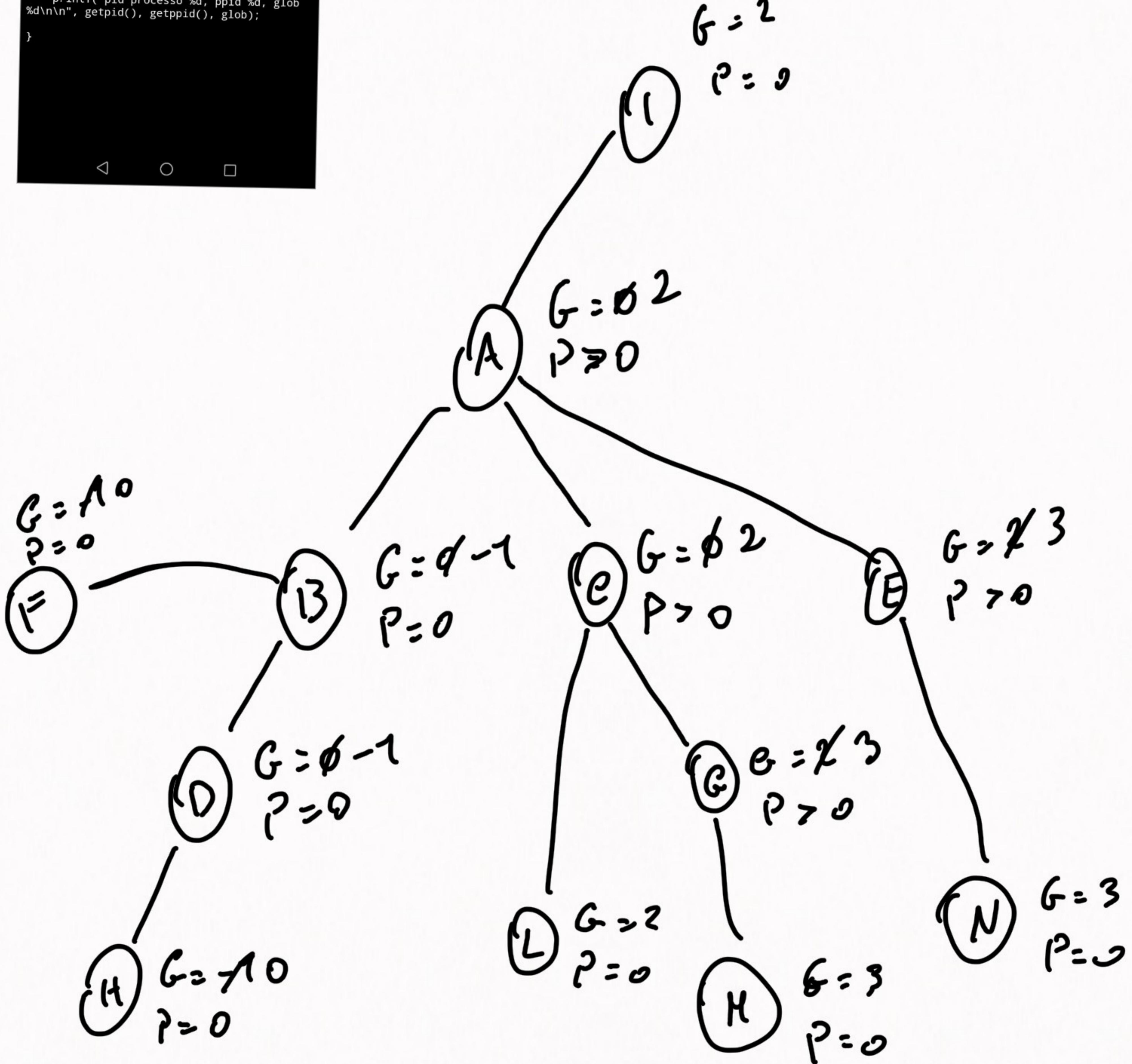
```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

int main()
{
    int glob = 0;
    int pid = 0;

    pid = fork();
    fork();

    if(pid)
        glob = 2;
    else
        glob--;
    if(!fork())
        glob++;
    for(int i = 1;i<2;i++)
    {
        if(glob>1)
            pid = fork();
        if(!pid)
            glob--;
        if(pid)
            glob--;
    }
    glob++;

    printf("pid processo %d, ppid %d, glob %d\n\n", getpid(), getppid(), glob);
}
```



vodafone IT 79% 13:04

```
← albero9Novembre.c :
```

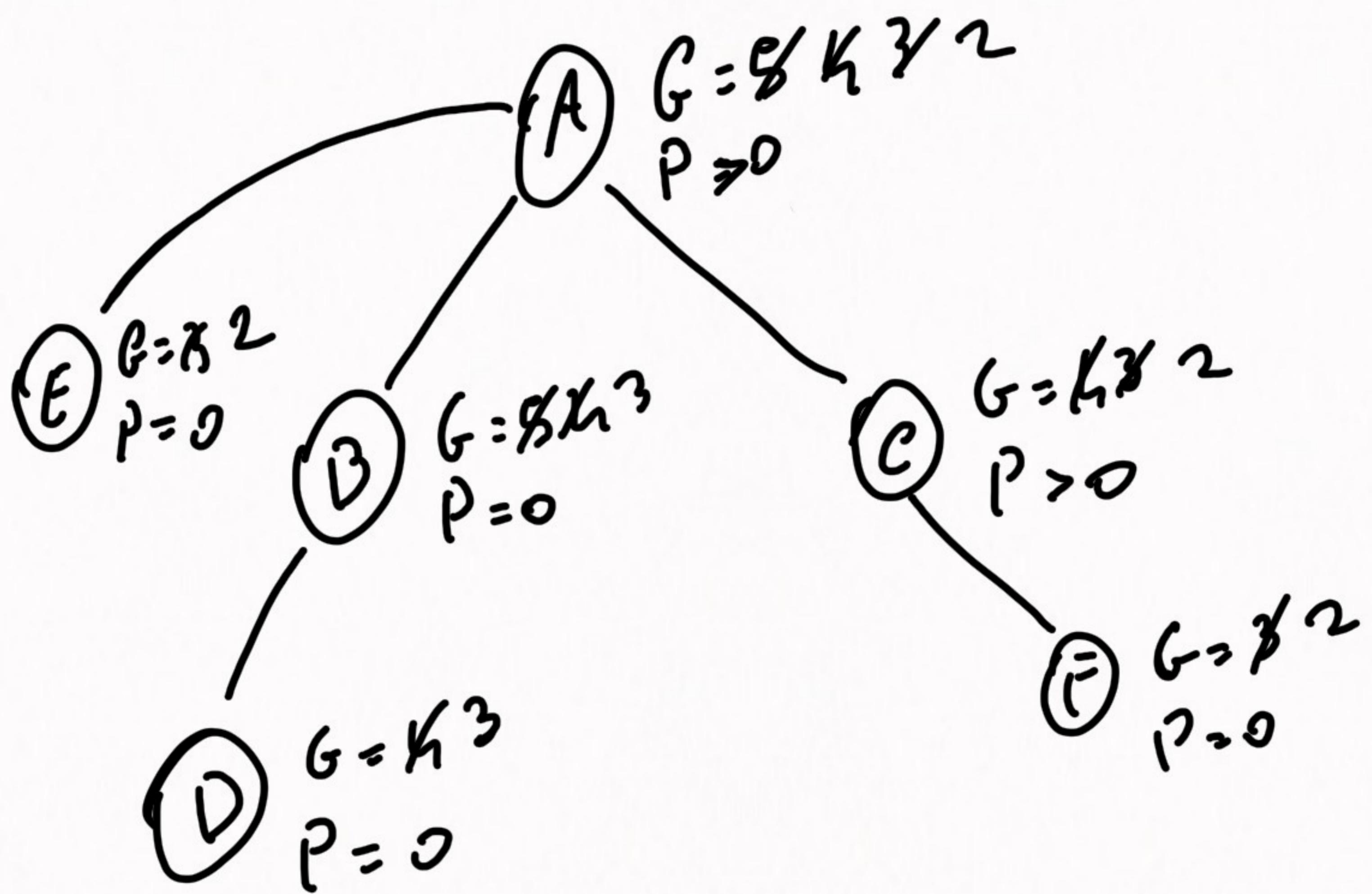
```
#include <unistd.h>
#include <stdio.h>

int main()
{
    int g = 5;
    int p = 0;

    p = fork();
    g--;
    fork();
    g--;

    if( p > 0 )
    {
        p = fork();
        g--;
    }

    printf("\n RIGA 23 pid = %d\nppid = %d\n\tglob=%d \tp=%d\n", getpid(), getppid(), g,
    p);
}
```



vodafone IT 79% 13:04

```
← albero31Luglio.c : 13:04
```

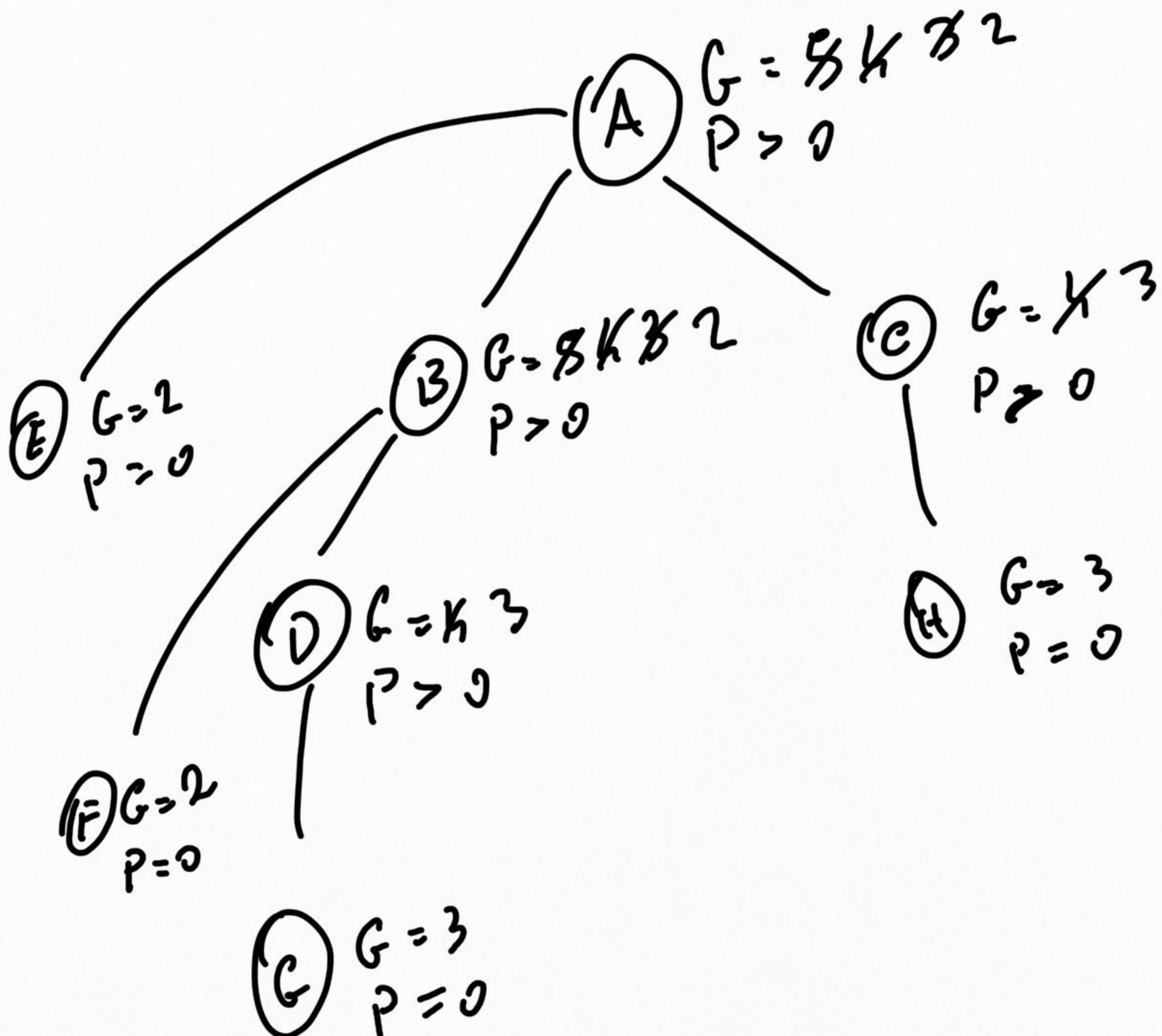
```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

int main()
{
    int glob = 5, pid = 0;

    fork();
    glob--;

    if( fork() )
        glob--;

    if( pid == 0 )
    {
        pid = fork();
        glob--;
    }
    printf("\n RIGA 24 pid = %d\tppid =
%d \tglob=%d\n\n", getpid(), getppid(),
glob);
}
```



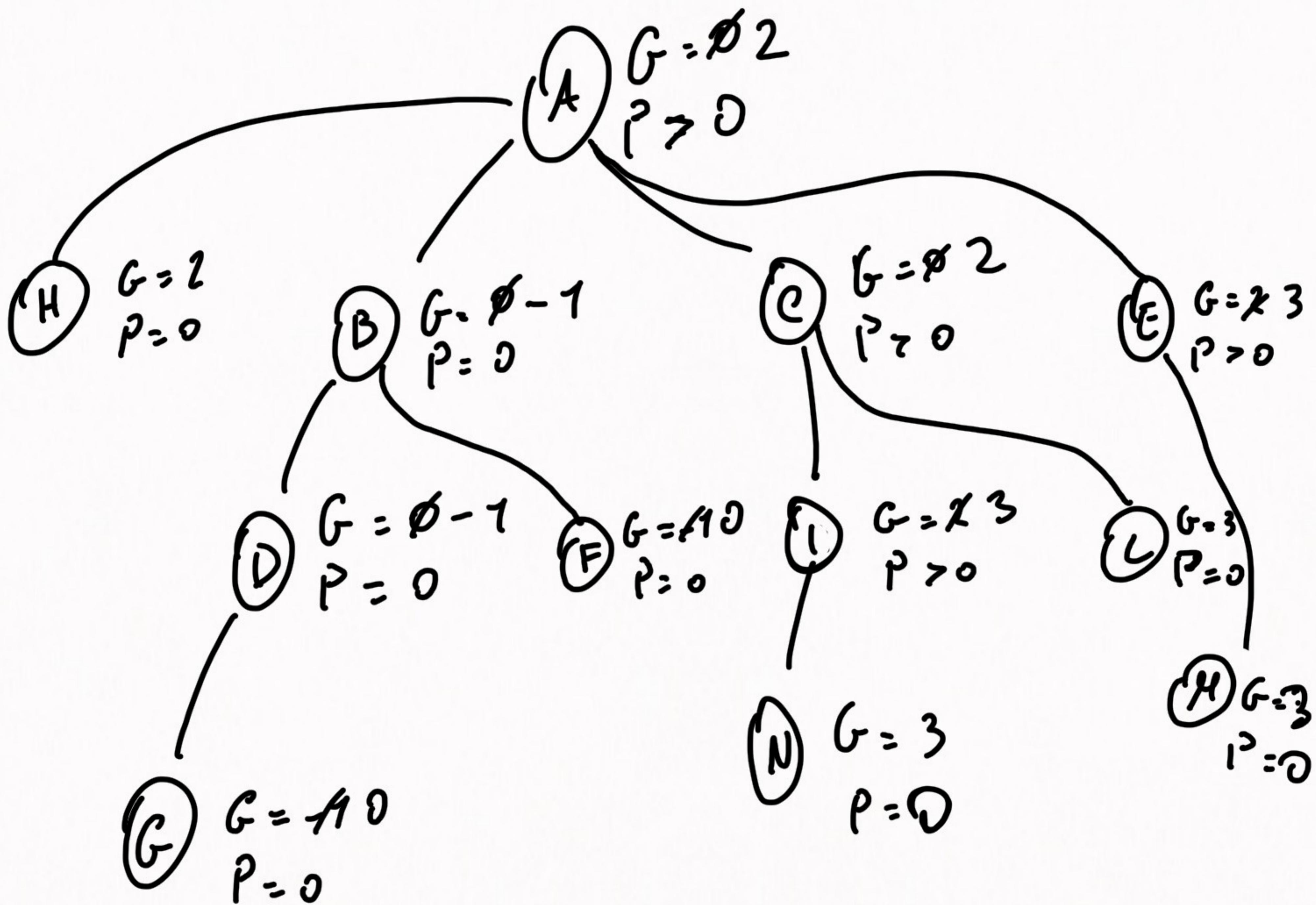
```

vodafone IT 79% 13:04
← alberello.c
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

int main()
{
    int glob = 0;
    int pid = 0;
    pid = fork();
    fork();
    if(pid)
        glob = 2;
    else
        glob--;
    if(!fork())
        glob++;
    for(int i = 1;i<2;i++)
    {
        if(glob>1)
            pid = fork();
        if(!pid)
            glob--;
        if(pid)
            glob--;
    }
    glob++;

    printf("pid processo %d, ppid %d, glob
%d\n\n", getpid(), getppid(), glob);
}

```



19:44 Mer 29 Mag 48% 79% 13:04

← utili\_so/WhatsApp Image 2023-09-14 at 13.05.34.jpeg

vodafone II

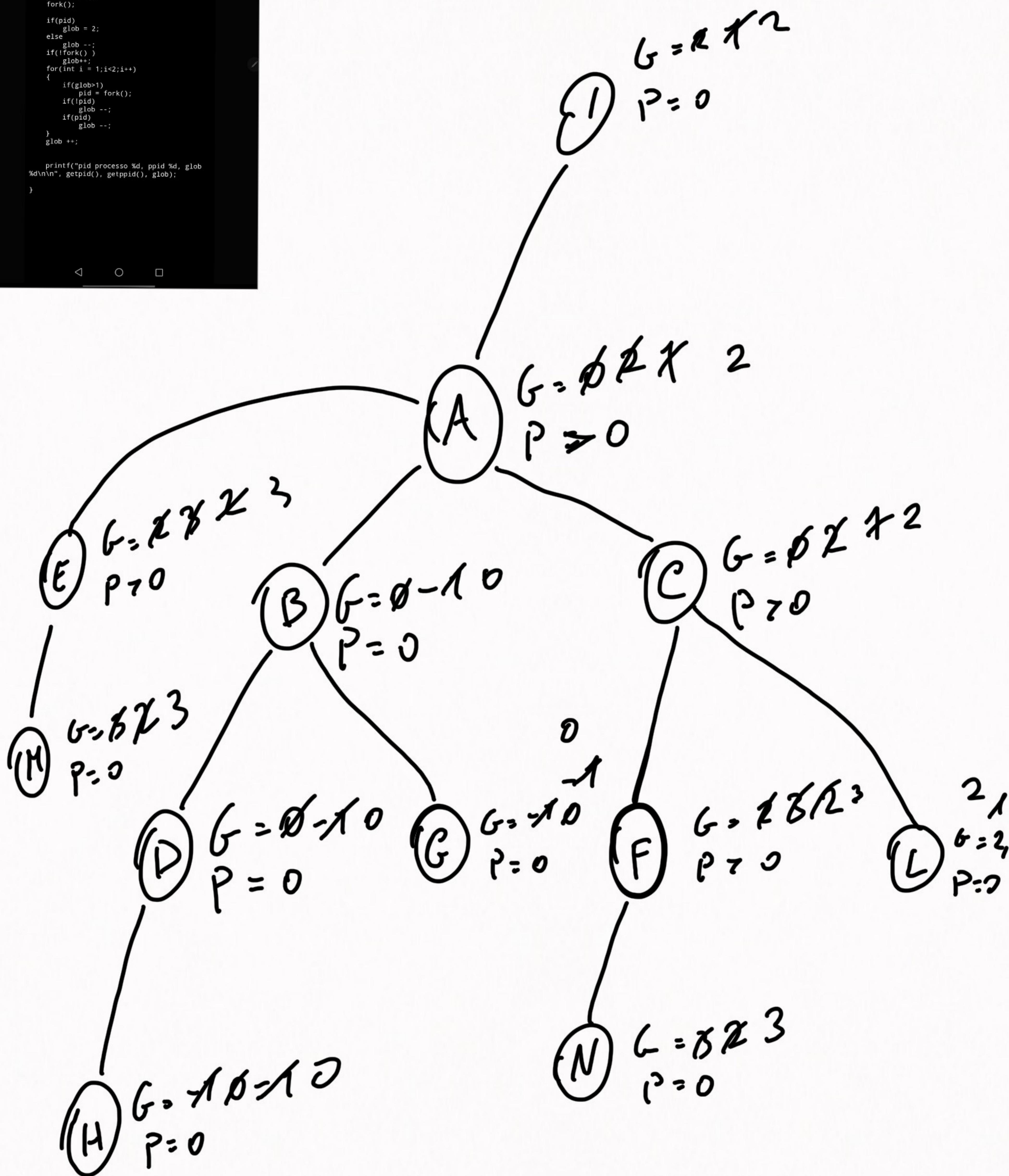
```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

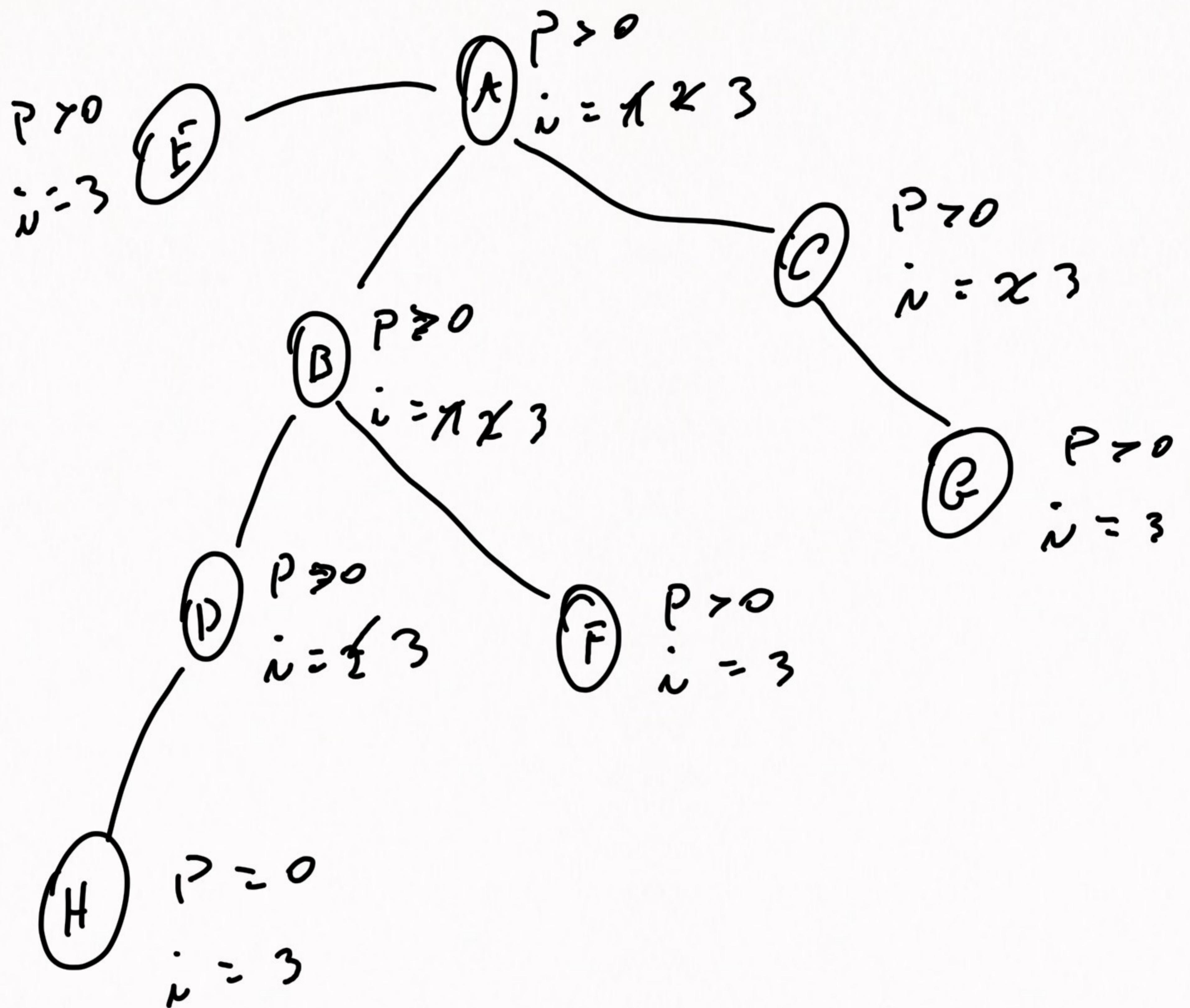
int main()
{
    int glob = 0;
    int pid = 0;

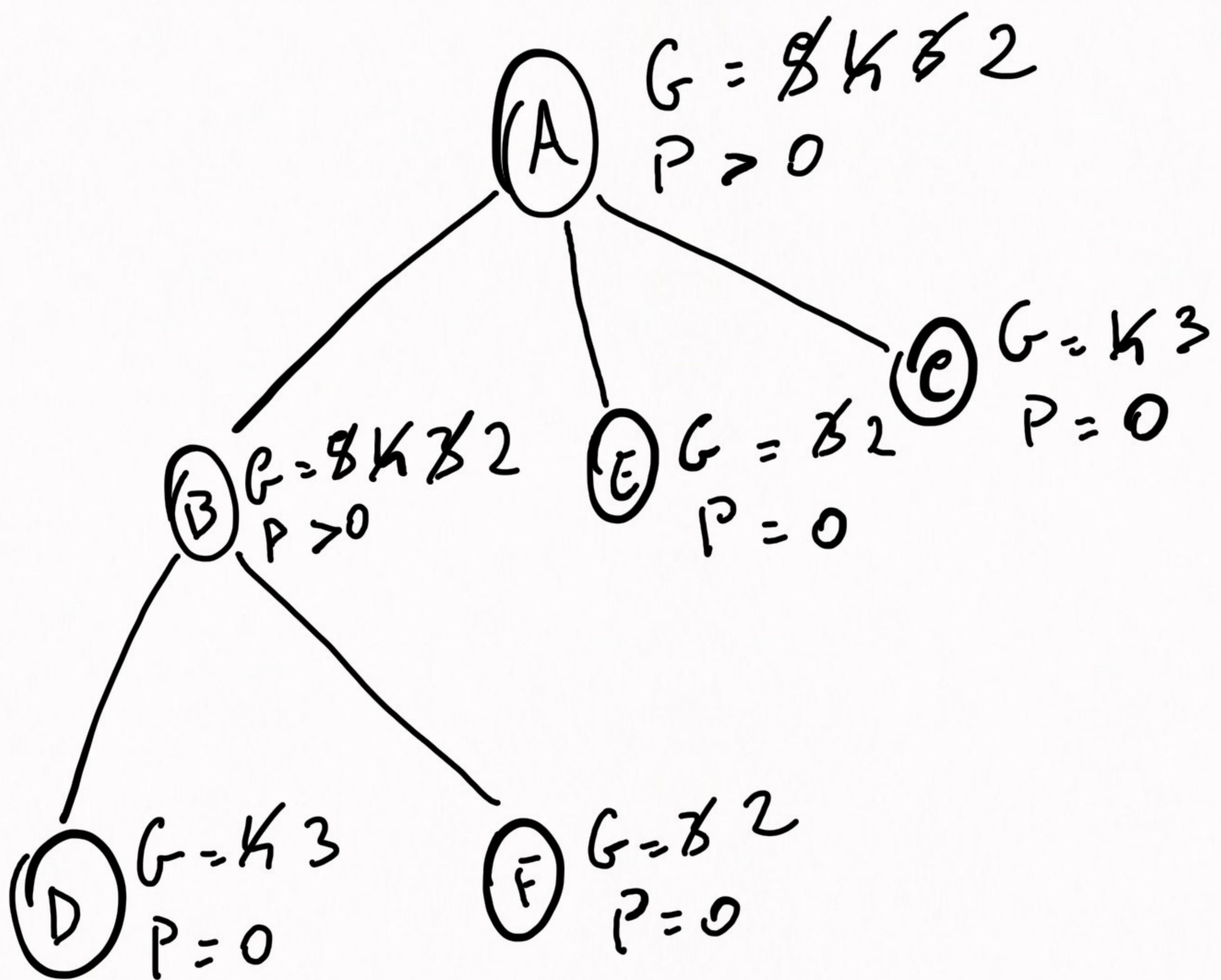
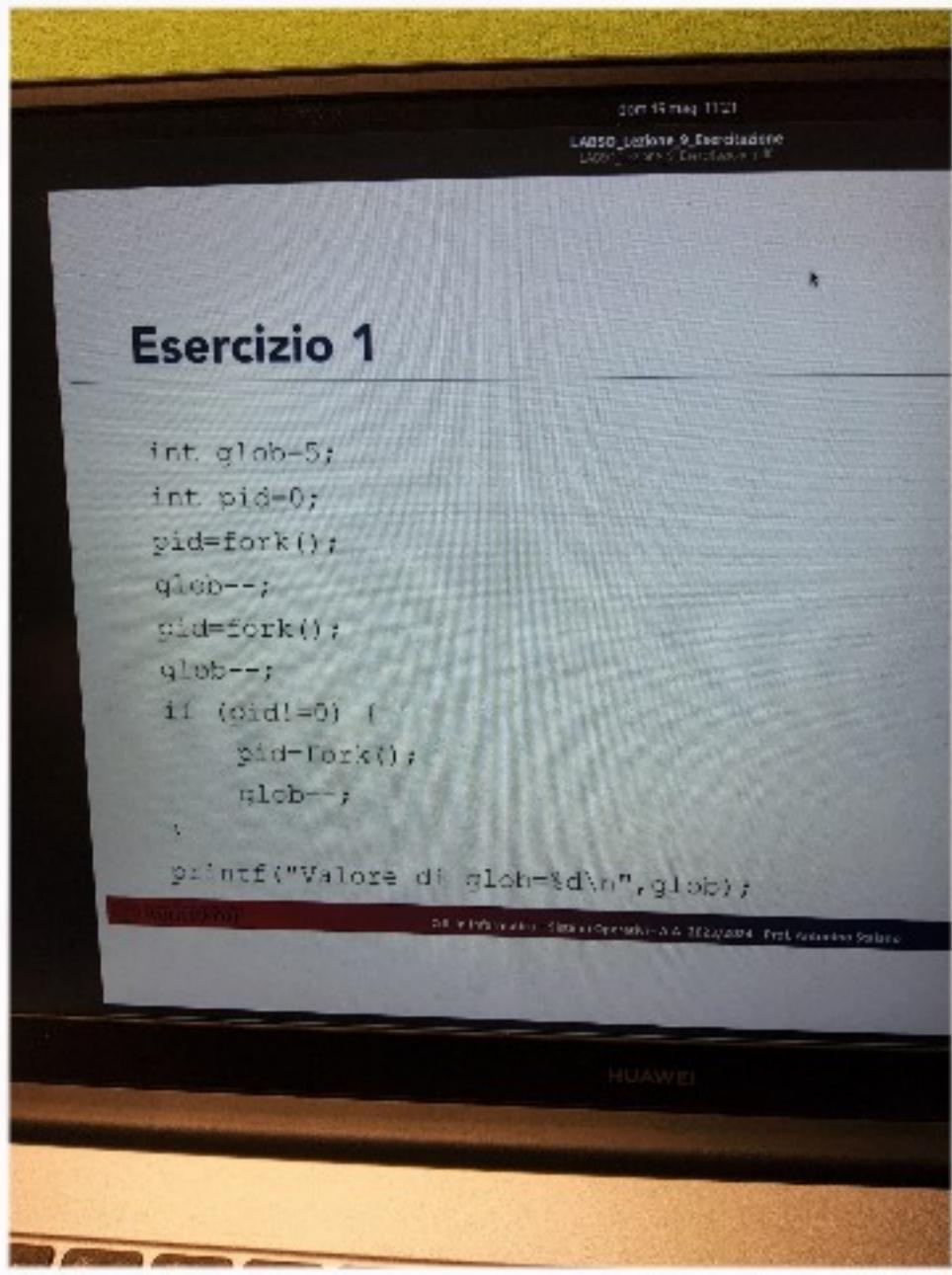
    pid = fork();
    fork();

    if(pid)
        glob = 2;
    else
        glob--;
    if(!fork())
        glob++;
    for(int i = 1; i<2; i++)
    {
        if(glob>1)
            pid = fork();
        if(pid)
            glob--;
        if(pid)
            glob--;
    }
    glob++;
}

printf("pid processo %d, ppid %d, glob
%d\n", getpid(), getppid(), glob);
}
```



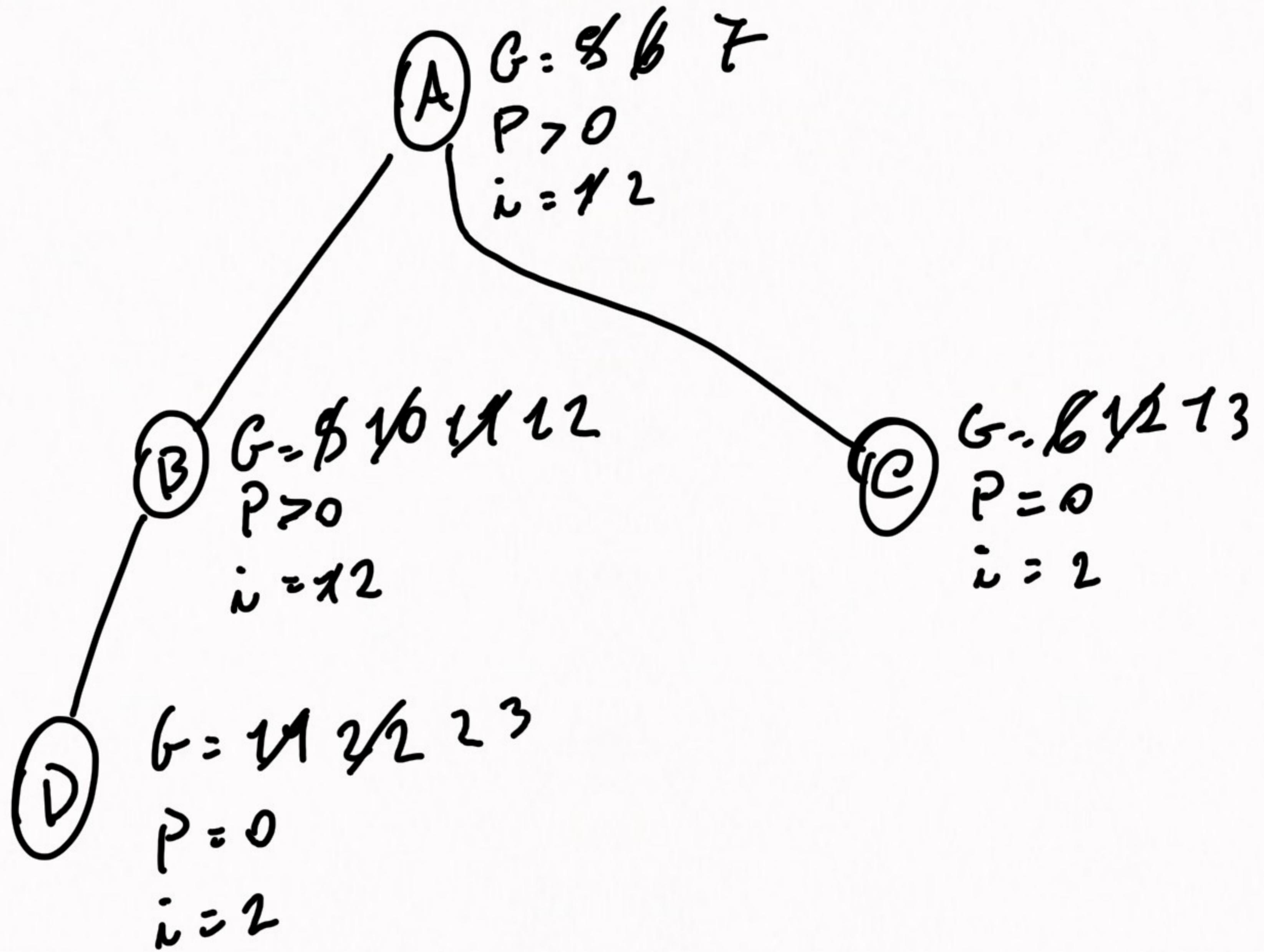




## Esercizio 2

```
int glob=5;
int pid=0;
int main() {
    int i=0;
    for (i=1;i<3;i++) {
        pid=fork();
        if (pid==0) {
            glob=glob*2;
            sleep(i+1);
        }
        glob=glob+5;
    }
    printf("Valore di glob=%d\n",glob);
}
```

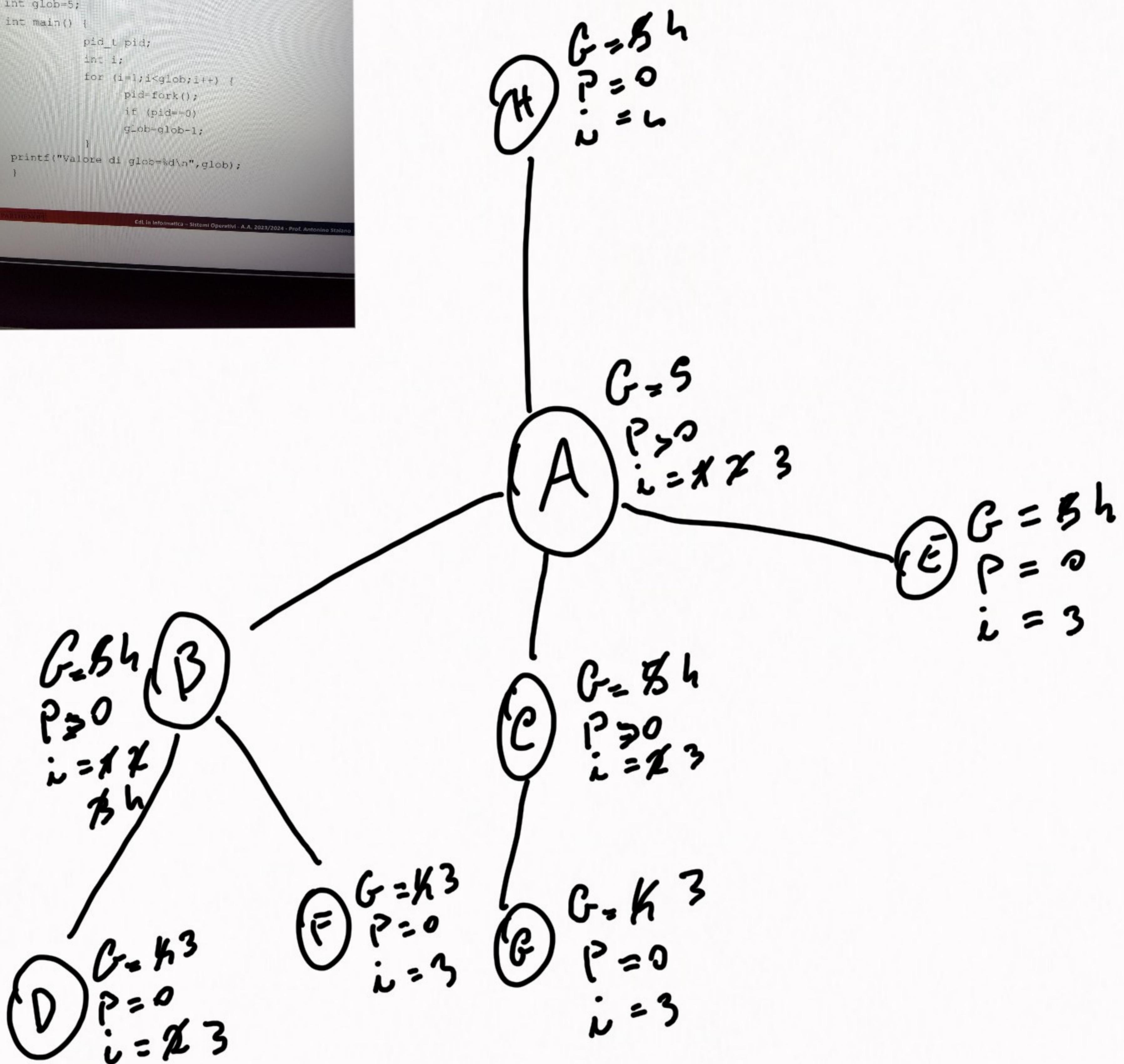
Cdl in Informatica - Sistemi Operativi - A.A. 2021/2022  
HUAWEI



## Esercizio

```
int glob=5;
int main() {
    pid_t pid;
    int i;
    for (i=1;i<glob;i++) {
        pid=fork();
        if (pid==0)
            glob=glob-1;
    }
    printf("Valore di glob=%d\n",glob);
}
```

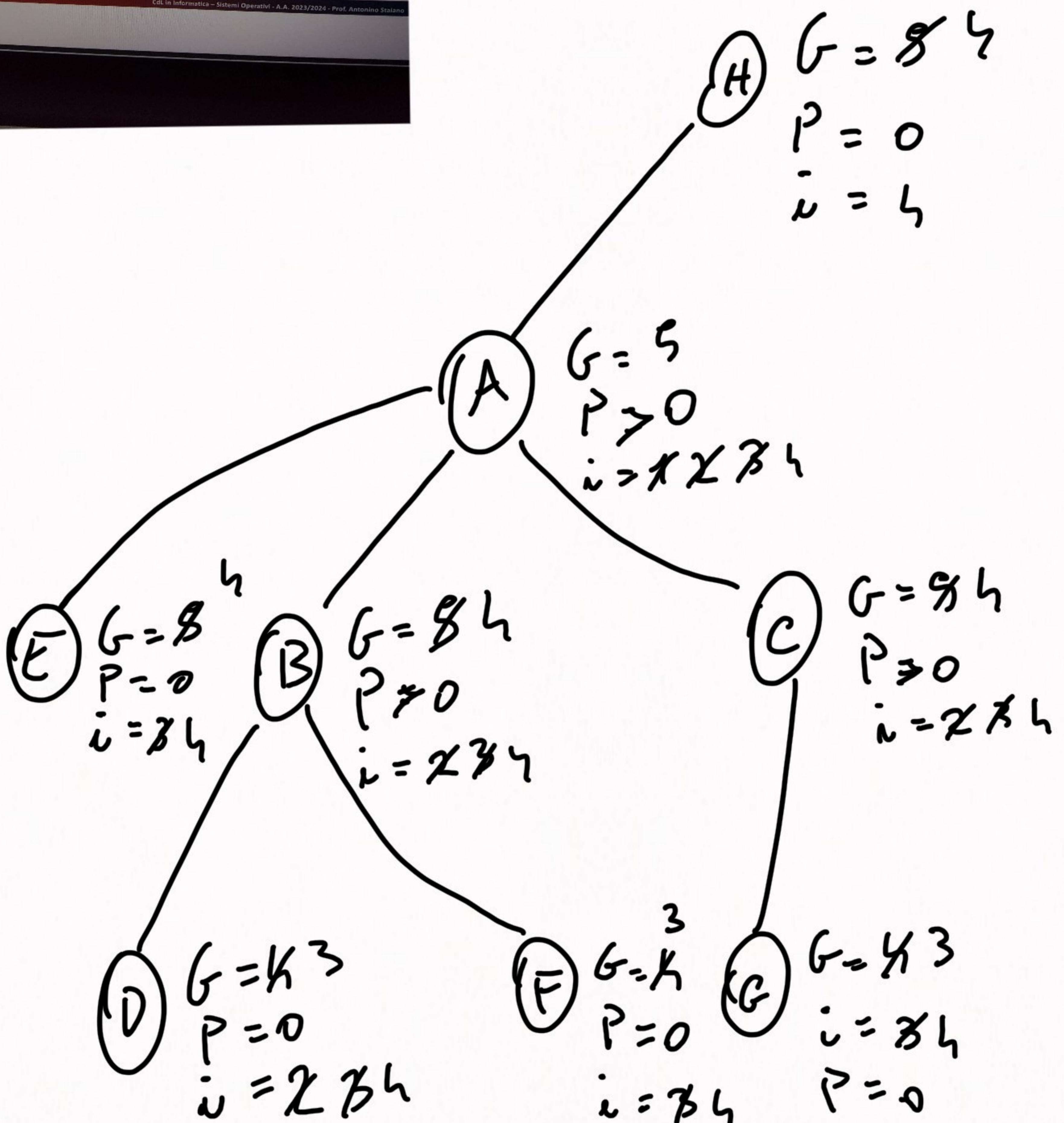
PARTIMENTO DI INGEGNERIA  
Ed. In Informatica - Sistemi Operativi - A.A. 2023/2024 - Prof. Antonino Stilo



## Esercizio

```
int glob=5;
int main() {
    pid_t pid;
    int i;
    for (i=1;i<glob;i++) {
        pid=fork();
        if (pid==0)
            glob=glob-1;
    }
    printf("Valore di glob=%d\n",glob);
}
```

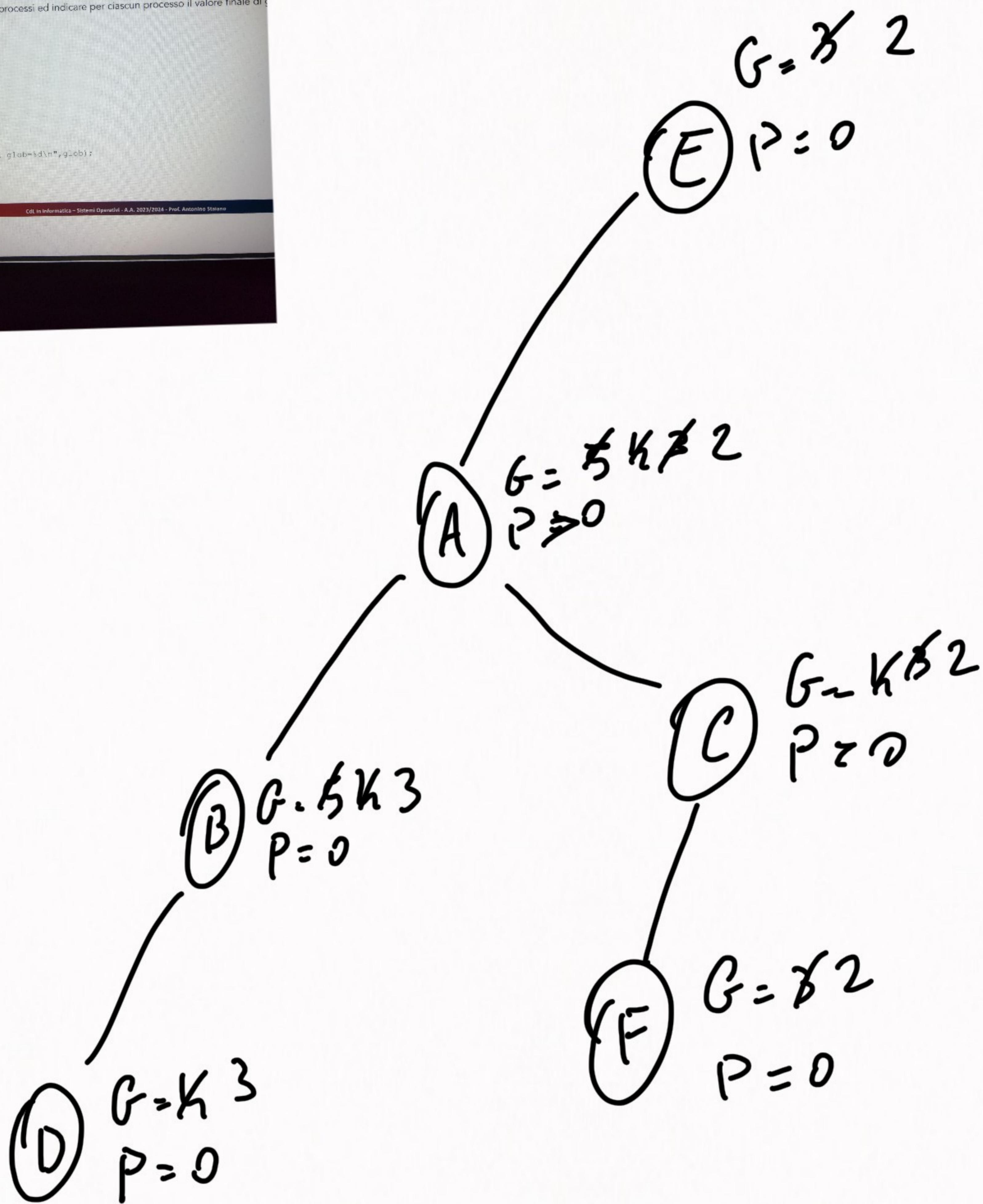
CdL in Informatica – Sistemi Operativi - A.A. 2023/2024 - Prof. Antonino Stolano



## Esercizio

Costruire l'albero dei processi ed indicare per ciascun processo il valore finale di  $g$  e  $p$ .

```
int glob=5;
int pid=0;
pid=fork();
glob--;
fork();
glob--;
if (pid>0) {
    pid=fork();
    glob--;
}
printf("Valore di glob=%d\n",glob);
```



## Esercizio 2

Descrivere l'albero dei processi creato con il segu

```
int glob=2;  
int pid=0;  
int main() {  
    for (i=1;i<3;i++) {  
        pid=fork();  
        if (pid==0) {  
            glob=glob*2;  
            pid=fork();  
        }  
        glob=glob+1;  
    }  
    printf("Valore di glob=%d\n",glob);  
}
```

CdL in Informatica – Sistemi Operativi – 1

Autore: D. Cicali

Versione: 1.0

Data: 10/09/2014

Autore: D. Cicali

Versione: 1.0

Data: 10/09/20

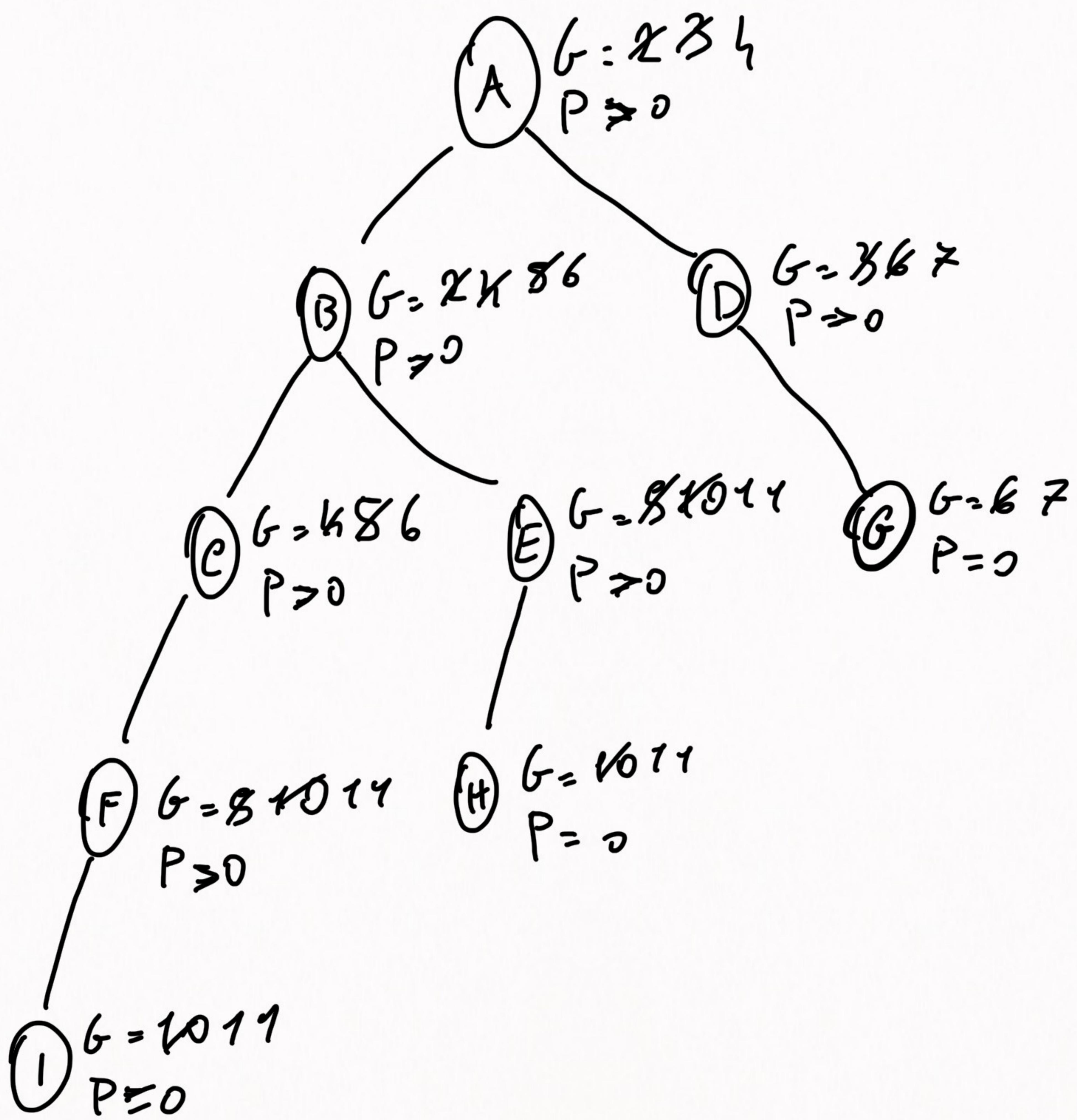
## Esercizio 2

Descrivere l'albero dei processi creato con il seguente codice.

```
int glob=2;  
int pid=0;  
int main() {  
    for (i=1;i<3;i++) {  
        pid=fork();  
        if (pid==0) {  
            glob=glob*2;  
            pid=fork();  
        }  
        glob=glob+1;  
    }  
    printf("Valore di glob=%d\n",glob);  
}
```

PARTHENOPE

CdL in Informatica - Sistemi Operativi -

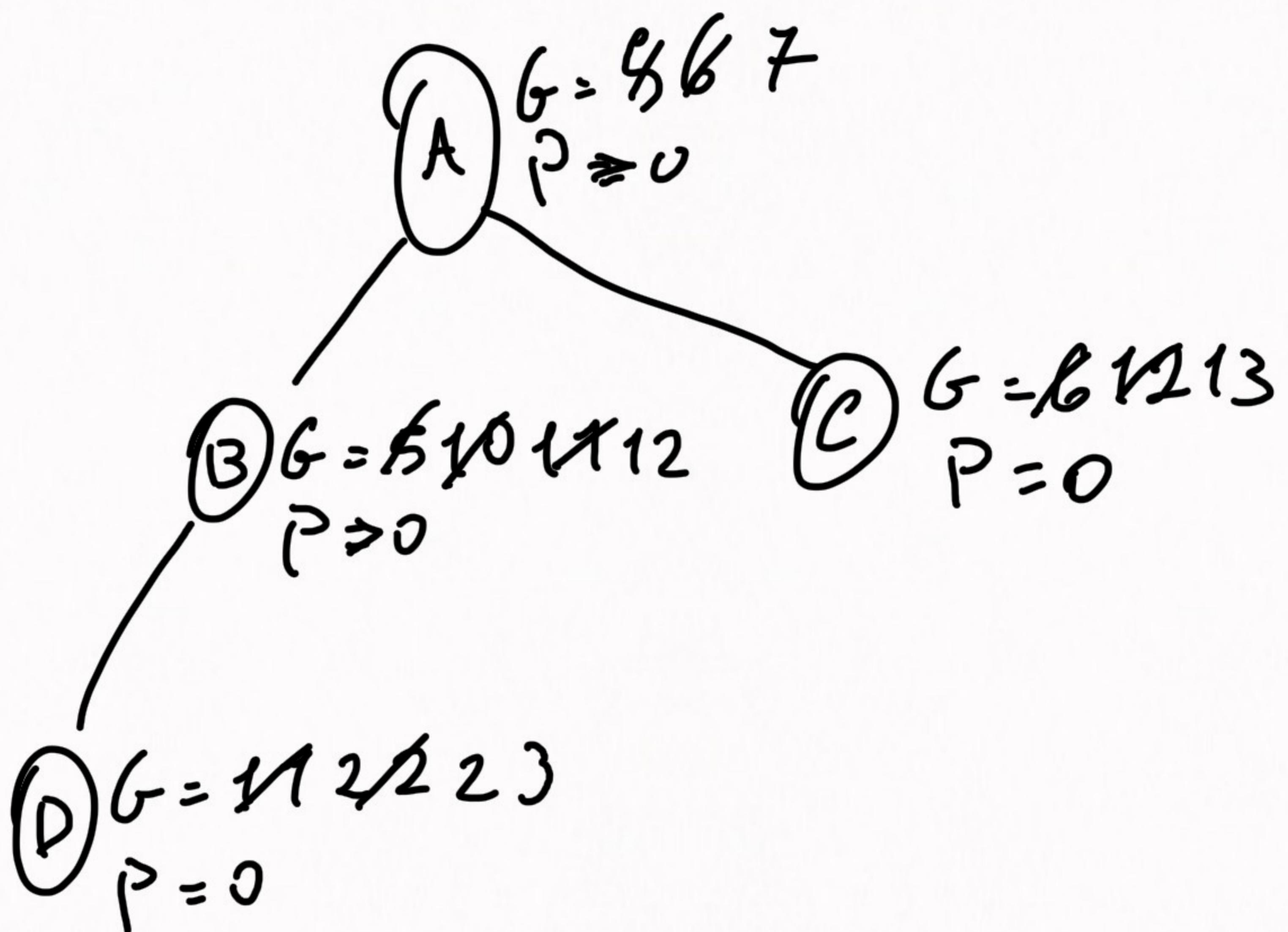


## Esercizio 2

```
int glob=5;
int pid=0;
int main() {
    int i=0;
    for (i=1;i<3;i++) {
        pid=fork();
        if (pid==0) {
            glob=glob*2;
            sleep(i+1);
        }
        glob=glob+1;
    }
    printf("Valore di glob=%d\n",glob);
}
```

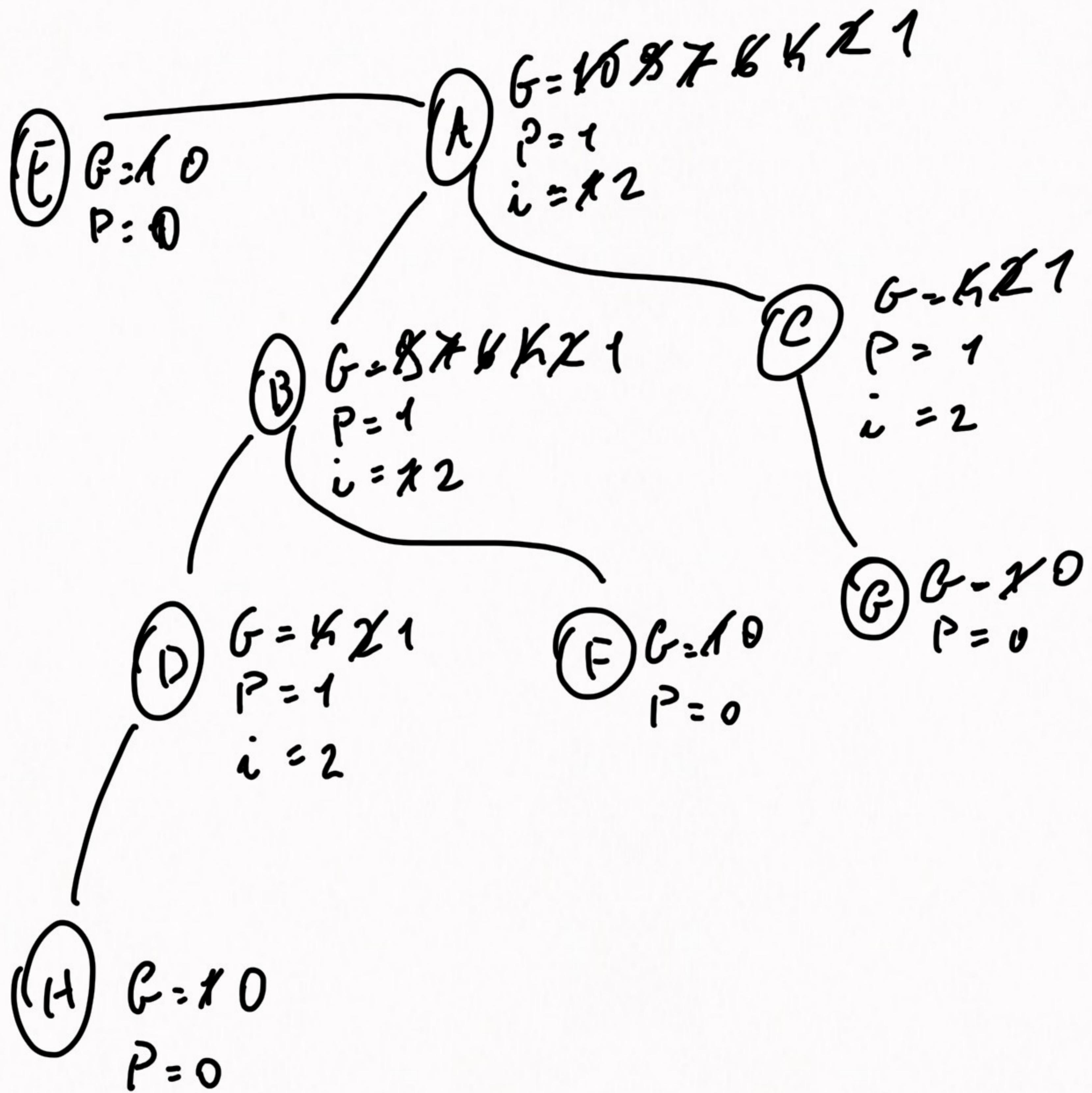
Cdl in Informatica - Sistemi Operativi - A.A. 2023/2024

HUAWEI



## Esercizio 5

```
int glob = 10;
int pid = 1;
int main(){
    for (i=1;i<3;i++) {
        glob = glob - i;
        fork();
        if (!pid)
            pid = fork();
        if (pid)
            glob-=2;
        glob-=1;
    }
    if (!(pid=fork()))
        glob--;
    printf("Valore di glob = %d\n",glob);
}
```

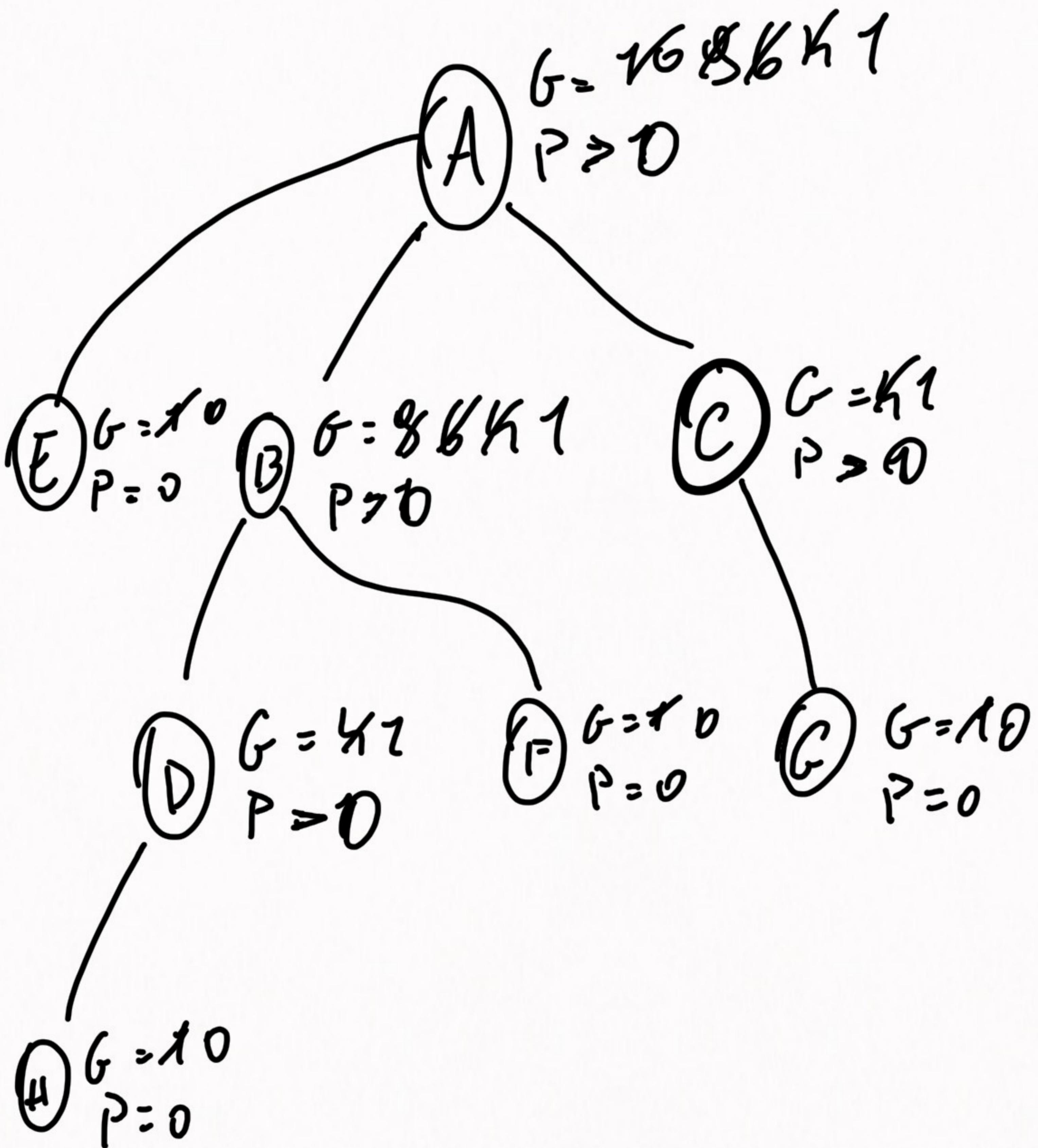


## Esercizio 5

```
int glob = 10;
int pid = 1;
int main(){
    for (i=1;i<3;i++){
        glob = glob - i;
        fork();
        if (!pid)
            pid = fork();
        if (pid)
            glob-=2;
        glob-=1;
    }
    if (!(pid=fork()))
        glob--;
    printf("Valore di glob = %d\n",glob);
}
```

PARTHENOPÉ

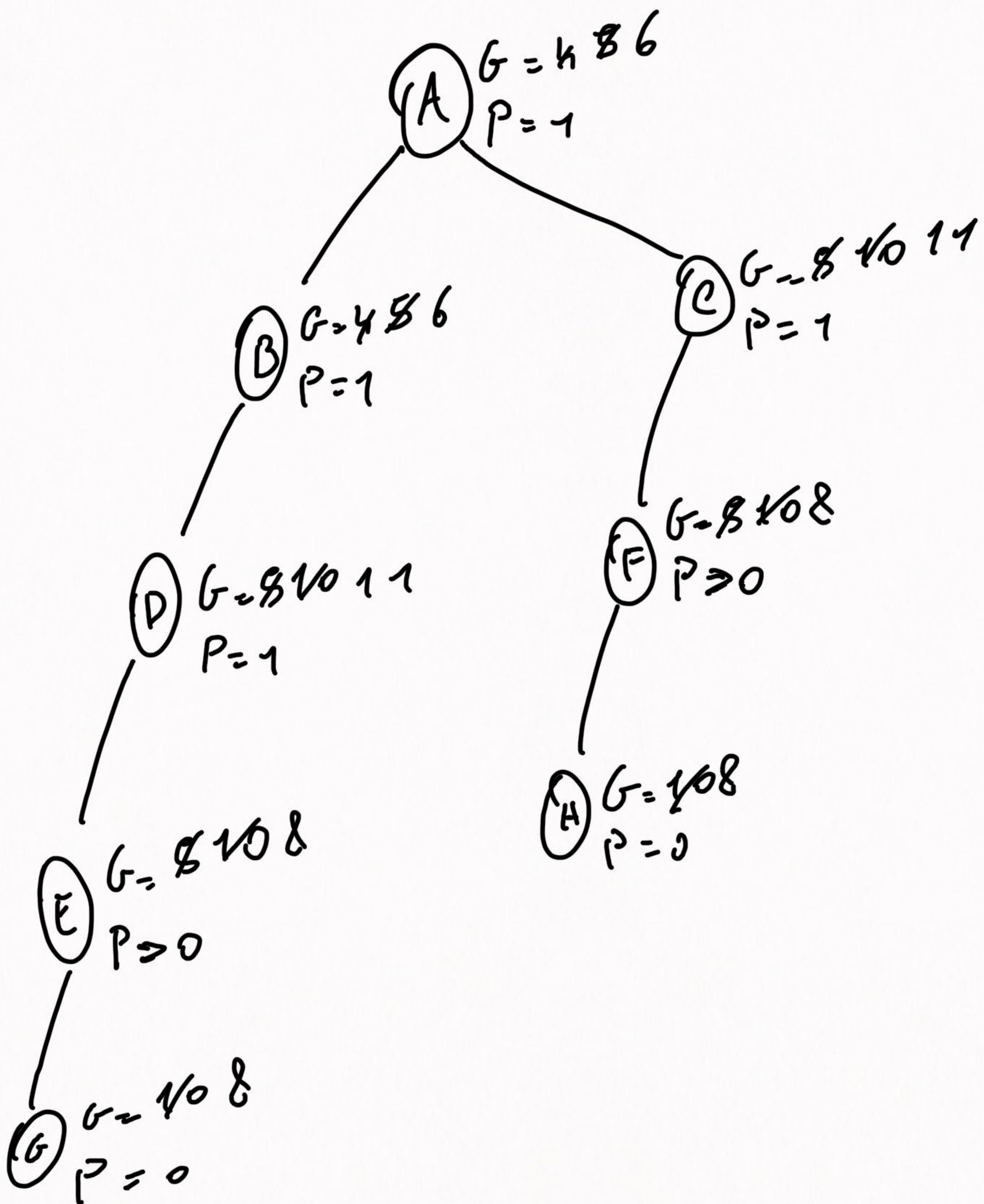
CdL in Informatica - Sistemi Operativi -



```

1. int glob=4;
2. int pid=1;
3. fork();
4. glob++;
5. if(!fork()) {
6.     pid=fork();
7.     glob=glob*2;
8. }
9. if (!pid)
10.    fork();
11.    glob=glob-2;
12. else
13.    glob++;
14. printf("Valore di glob=%d\n",glob);

```

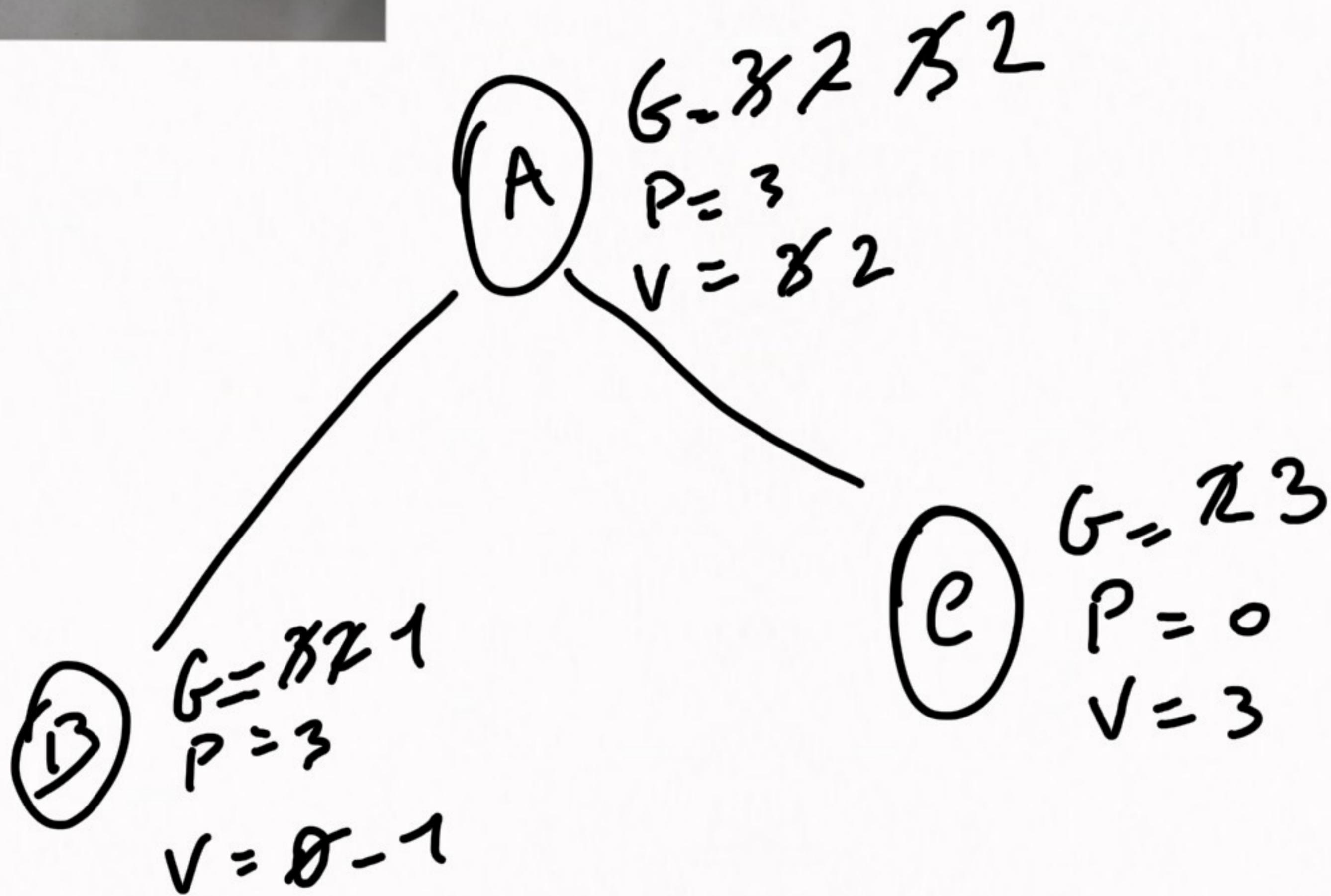


MARIO VISTA 0124001266

Costruire l'albero dei processi ed indicare per ciascun processo il valore finale delle variabili `glob`, `pid` e `var`.

```

1. int glob=3;
2. int pid=3;
3. int var=3;
4.
5. int main(){
6.     var=fork();
7.     glob--;
8.
9.     if (var){
10.         pid=fork();
11.         glob++;
12.     }
13.     if (pid){
14.         glob--;
15.         var--;
16.     }
17.     if (!glob){
18.         var=fork();
19.         pid++;
20.     }
21. }
22. printf("Valore di glob=%d\n",glob);
23. printf("Valore di pid=%d\n",pid);
24. printf("Valore di var=%d\n",var);
25.
26.
27. }
```



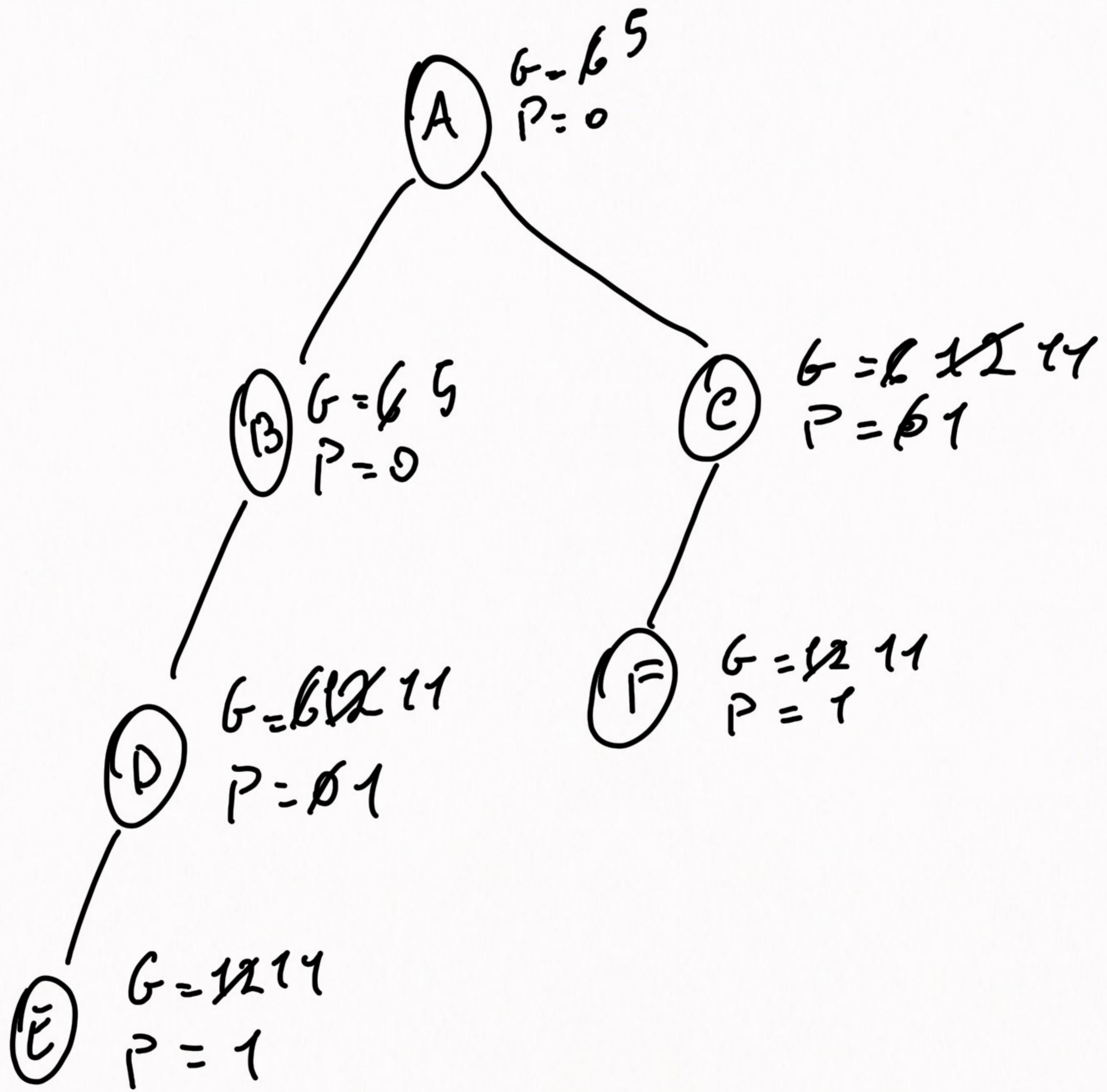
IACOMINO Giuse

Costruire l'albero dei processi ed indicare per ciascun processo il valore finale della variabile glob

```

int glob=6;
int pid=0;
fork();
for (i=1; i<2; i++) {
    if(!fork()){
        glob=glob*2;
        pid=1;
    }
    if(pid)
        fork();
    glob=glob-1;
}
printf("Valore di glob=%d\n",glob);
    
```

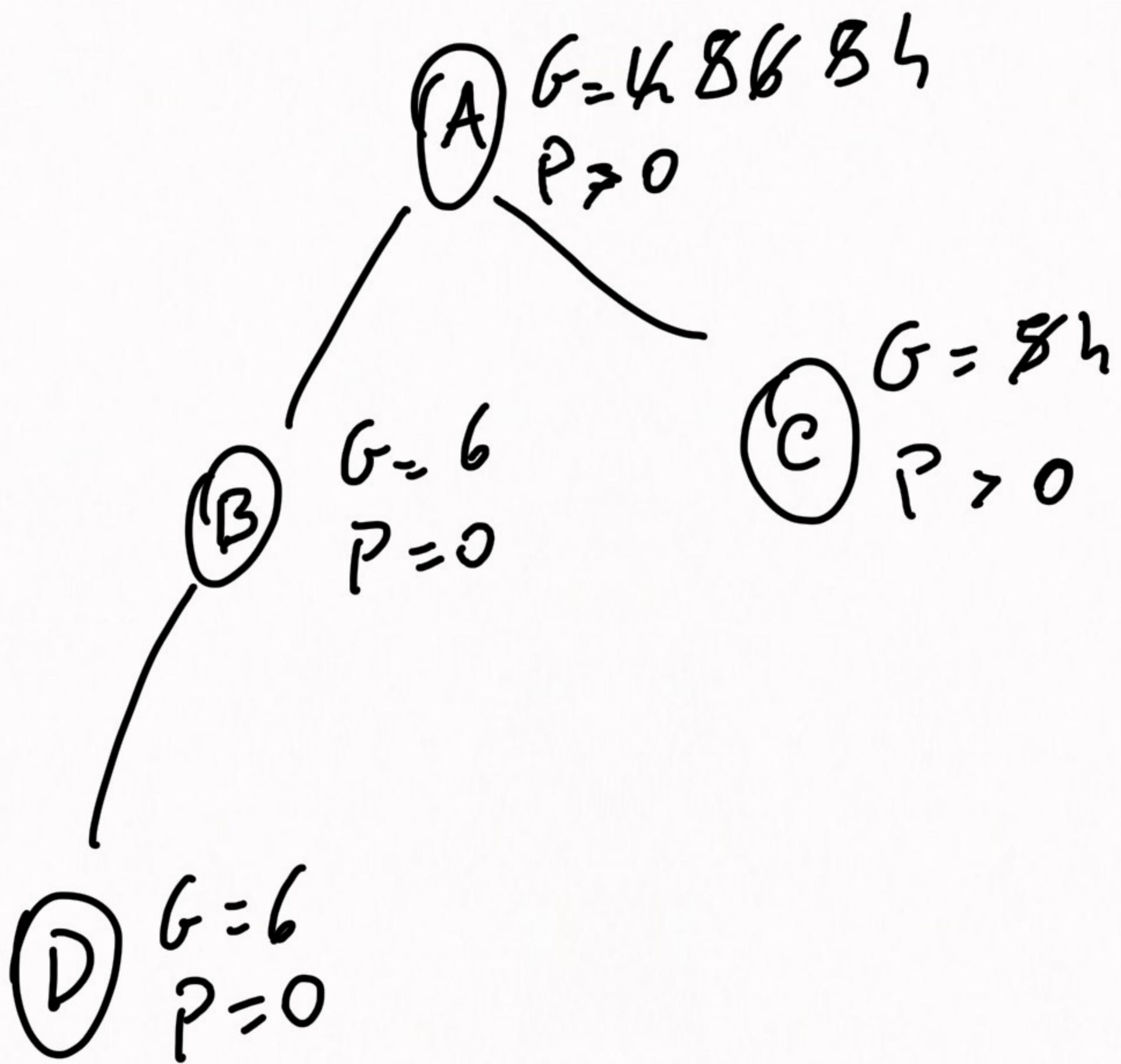
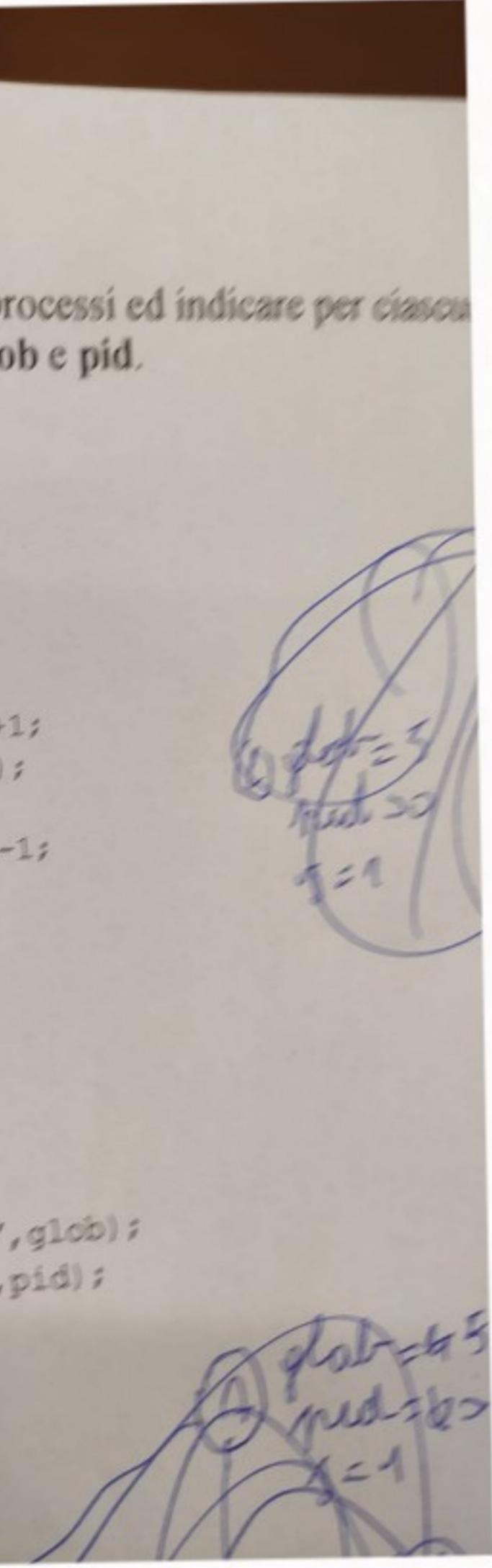
(A)  $\overset{g=6}{p=0}$



Costruire l'albero dei processi ed indicare per ciascuna finale delle variabili glob e pid.

```
int glob=4;
int pid=0;
int main()
{
    for (j=1;j<2;j++) {
        glob=glob+j;
        if (!pid)
            glob=glob+1;
        pid=fork();
        if (pid)
            glob=glob-1;
        fork();
    }

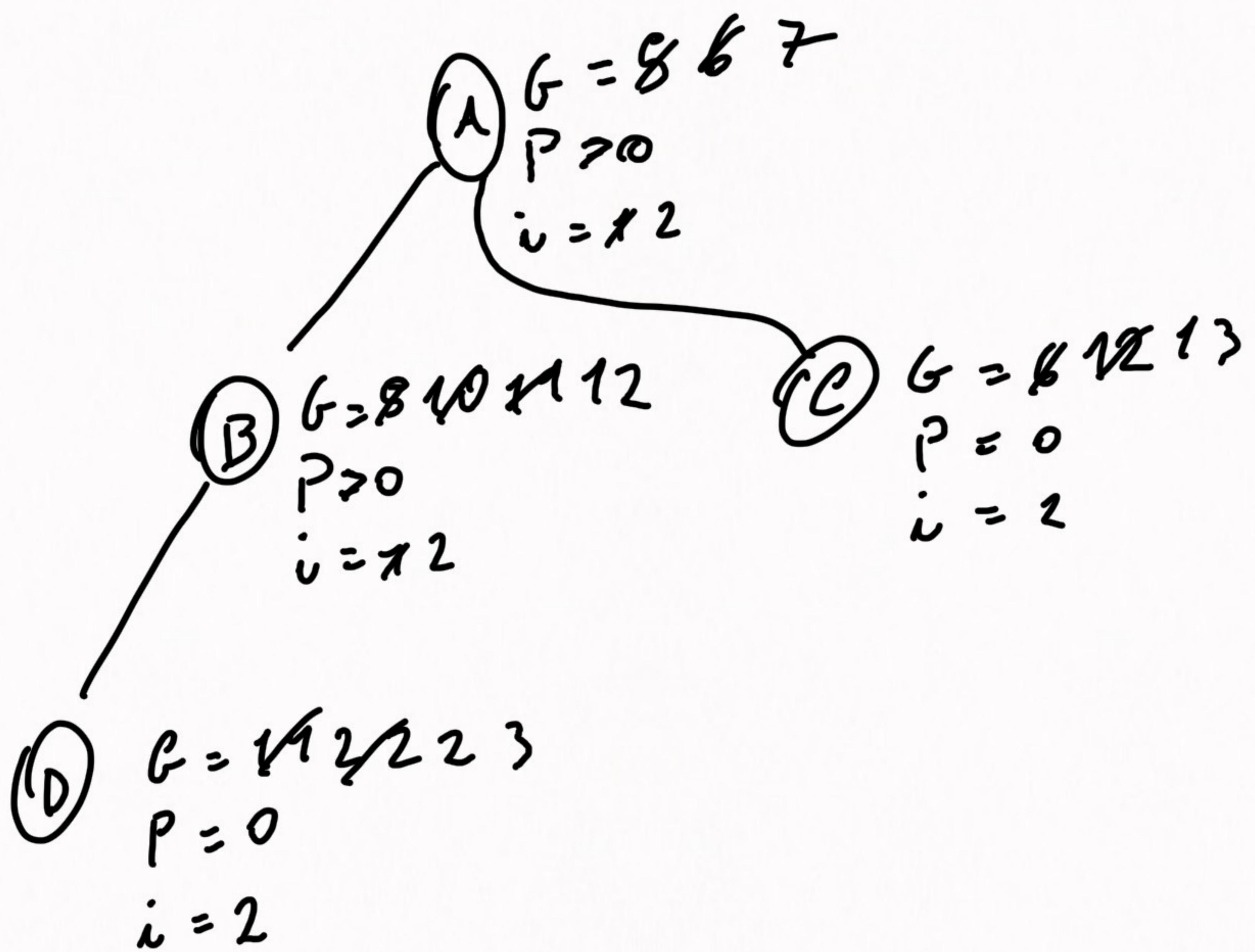
    if (!pid)
        fork();
    if (pid)
        glob--;
    printf("glob=%d\n",glob);
    printf("pid=%d\n",pid);
}
```



## Esercizio 3

Costruire l'albero dei processi ed indicare per ciascun processo il valore finale della variabile glob

```
int glob=5;
int pid=0;
int main() {
int i=0;
for (i=1;i<3;i++) {
pid=fork();
if (pid==0) {
glob=glob*2;
sleep(i+1);
}
glob=glob+1;
printf("Valore di glob=%d\n",glob);
}
```



Video Prog Varie Importati UNI Moto WhatsApp Ti diamo il ben... Posta in arrivo

4\_2024\_Esercitazione 2 / 6 - 100% + -

## Esercizio 1

Costruire l'albero dei processi ed indicare per ciascun proc

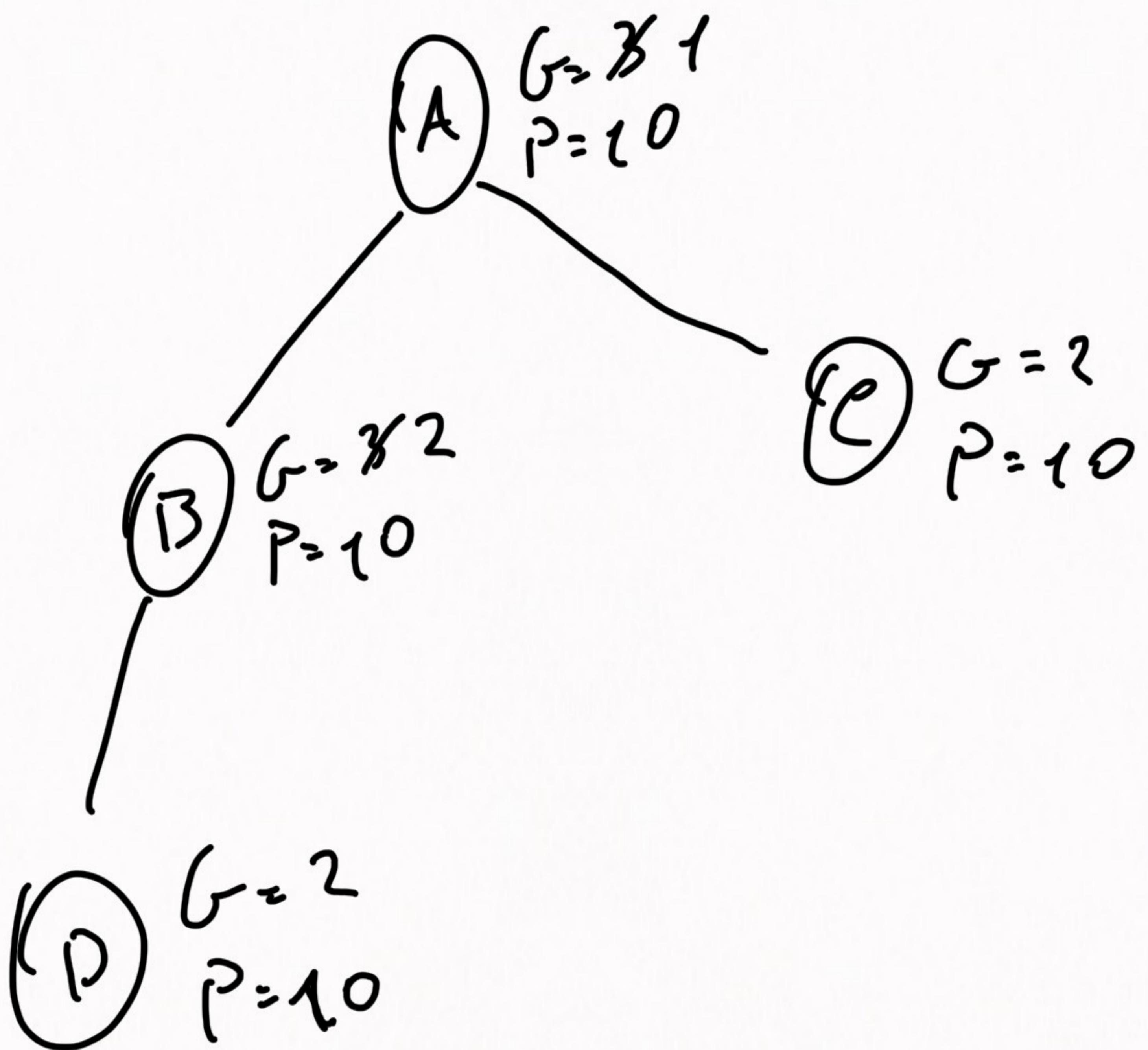
```

int glob = 3;
int pid = 10;
if (fork() == 0) {
    glob -= 2;
} else {
    glob--;
    if (!fork()) {
        if (pid > 0) {
            pid = fork();
            glob--;
        }
    }
}
printf("Valore di glob=%d\n", glob);

```

PARTHENOPE CdL in Informatica – Sistemi Operativi - A.A. 2023/2024 - Prof. A.

HUAWEI



## Esercizio 2

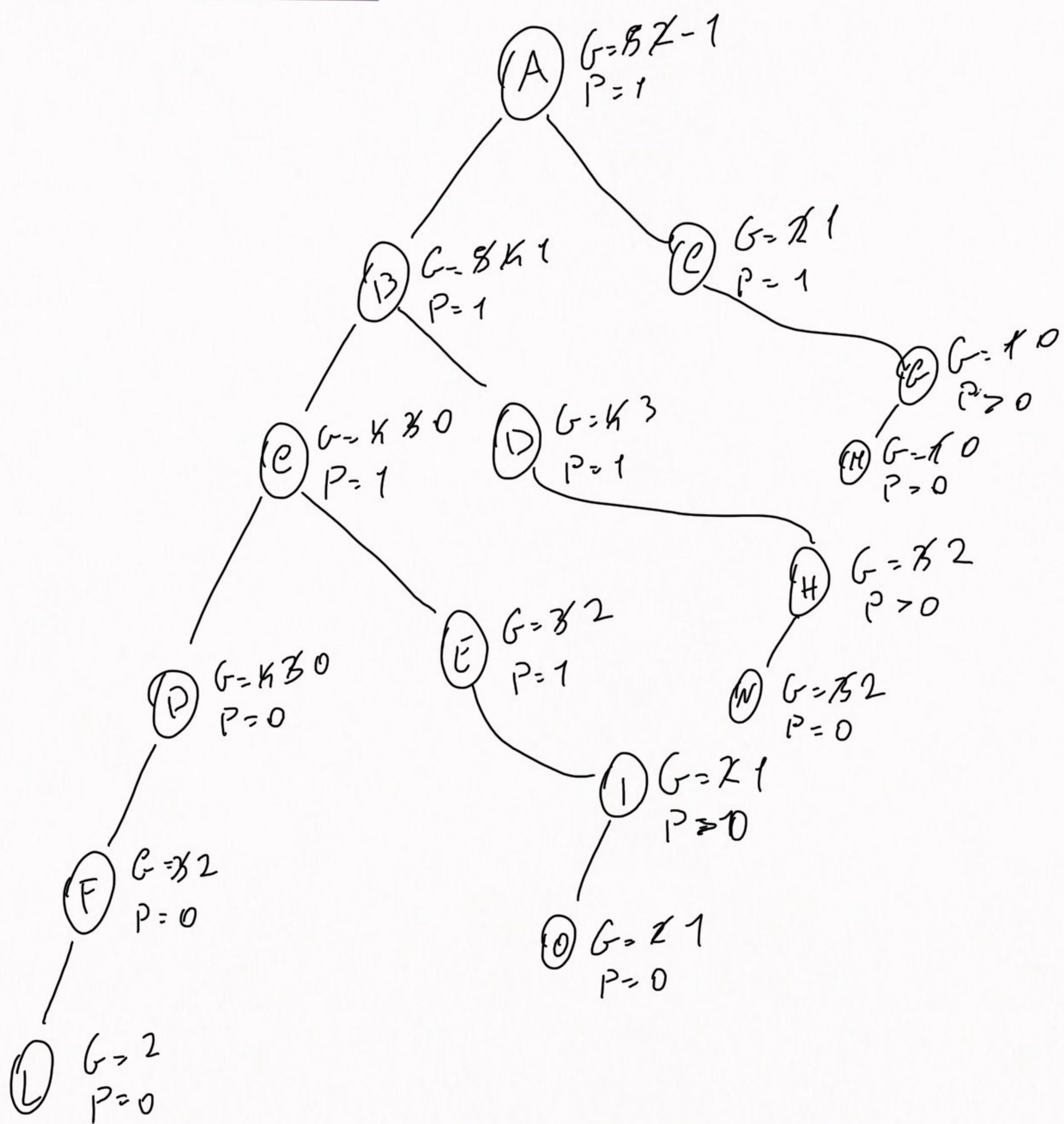
Descrivere l'albero dei processi creato con il seguente codice.

```

int glob = 5;
int pid = 1;
int counter = 2;
while (counter > 0) {
    if (fork()) {
        glob -= 3;
    } else {
        glob--;
        if (fork() == 0)
            if (pid > 0) {
                pid = fork();
                glob--;
            }
        counter--;
    }
}
printf("Valore di glob=%d\n", glob);

```

$$C = \Sigma 1$$



## ESERCIZIO 1

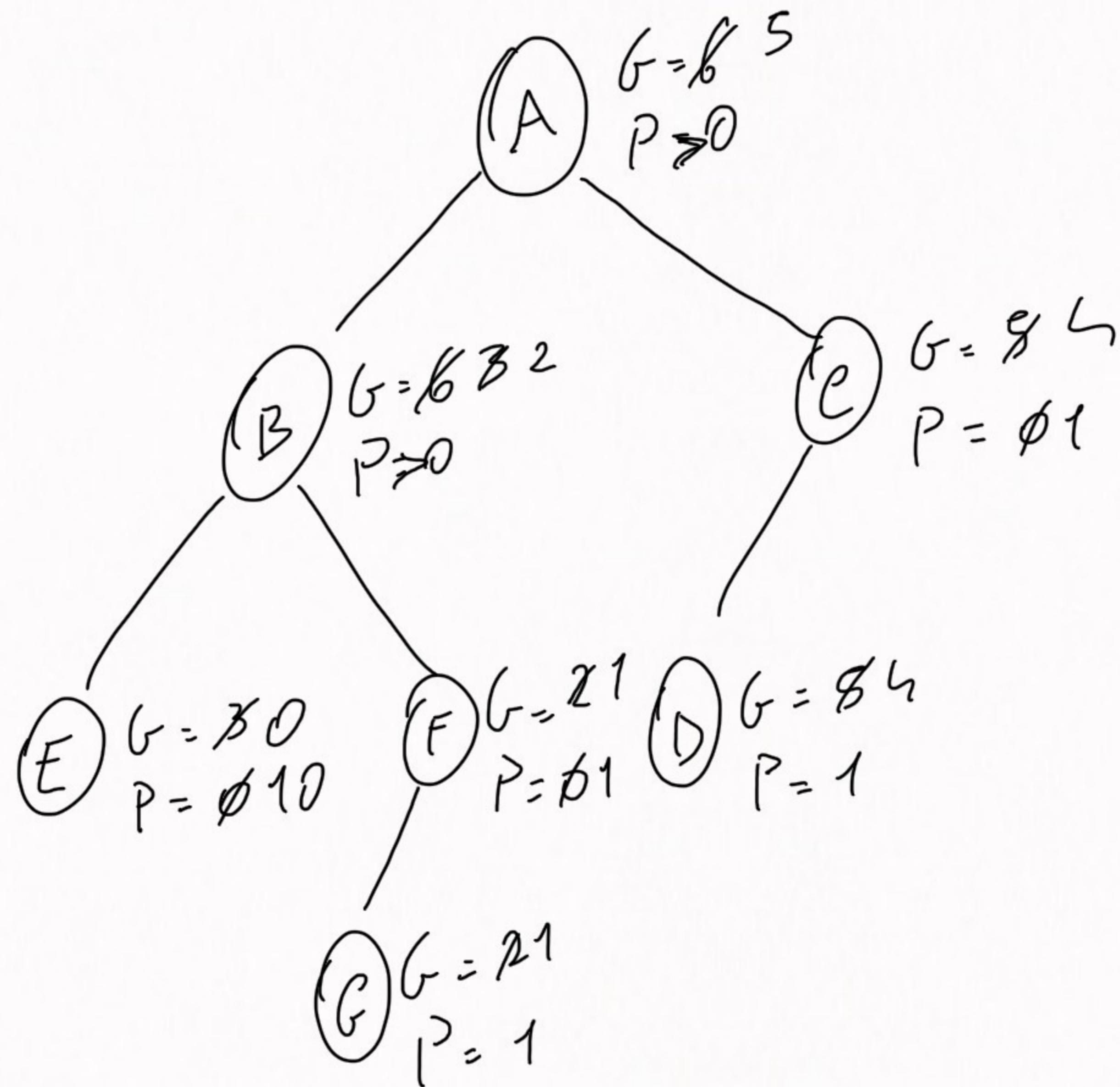
Costruire l'albero dei processi ed indicare per ciascuno il suo stato.

```
glob = 6;
pid = 0;
while (!pid) {
    if (!fork())
        glob -= 3;
    else {
        glob--;
        if ((pid=fork()) == 0) {
            pid =1
            fork();
            glob--;
        }
    }
    if (!glob)
        pid=10
}
printf("Valore di glob=%d\n", glob);
```

CdL in Informatica - Sistemi Operativi - A

PARTE NOPE

EDUCAVIT



vodafone IT 4G 79% 13:04

alberello.c

```
#include <stdio.h>
#include <unistd.h>
#include <stdlib.h>

int main()
{
    int glob = 0;
    int pid = 0;

    pid = fork();
    fork();

    if(pid)
        glob = 2;
    else
        glob--;
    if(!fork())
        glob++;
    for(int i = 1;i<2;i++)
    {
        if(glob>1)
            pid = fork();
        if(!pid)
            glob--;
        if(pid)
            glob--;
    }
    glob++;
}

printf("pid processo %d, ppid %d, glob
%d\n\n", getpid(), getppid(), glob);
}
```

