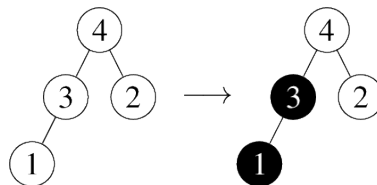


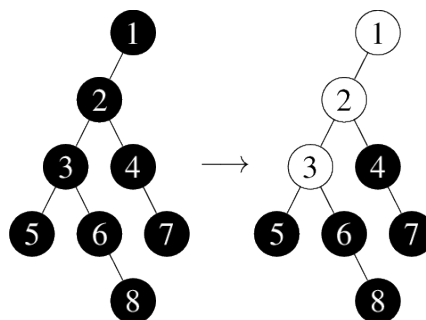
## Bollmaskinen

Vi har en “bollmaskin” som kan visualiseras som ett träd där en nod är rot. Trädets noder är numrerade från 1 till  $N$ . Varje nod är antingen tom eller innehåller en boll. Från början är alla noder tomma. När den kör kan maskinen utföra två sorters operationer:

1. Lägg till  $k$  bollar i bollmaskinen: Bollarna läggs en efter en i rotnoden. Så länge en boll har minst en tom nod direkt under sig (en barn-nod), så kommer den rulla nedåt till en av dessa. Om det finns flera tomma barn-noder så kommer bollen att välja den vars subträd innehåller lägst nod-nummer. Om bollen rullar ner flera nivåer, så gör den ett nytt val varje gång. Exempelvis, om vi lägger till två bollar i maskinen i figuren nedan, så hamnar de i noderna 1 och 3: Den första rullar från nod 4 till nod 3 eftersom nod 3 är tom och innehåller nod 1 i sitt subträd (vilket består av noderna 3 och 1); därefter rullar den vidare från nod 3 till nod 1. Den andra bollen rullar också från nod 4 till nod 3 och stannar där.



2. Ta bort en boll från en given nod: Denna nod blir tom och eventuella bollar uppifrån rullar ner. Närhelst en förälder till en tom nod innehåller en boll så kommer denna boll att rulla ner. Om vi exempelvis tar bort bollar från noderna 5, 7 och 8 (i denna ordning) från maskinen i bilden nedan, så kommer noderna 1, 2 och 3 att bli tomma.



## Indata

Första raden innehåller två heltal  $N$  och  $Q$  – antalet noder i trädets och antalet operationer. De följande  $N$  raderna beskriver bollmaskinen. Var och en av dessa rader innehåller ett heltal: den  $i$ -te av dessa rader innehåller nod-numret för nod  $i$ 's föräldernod, eller 0 om nod  $i$  är trädets rot. Sedan följer  $Q$  rader, som vardera innehåller två heltal och beskriver en operation som ska utföras. En operation av första typen beskrivs med  $1 k$  där  $k$  är antalet bollar som ska läggas till i maskinen. En operation av den andra typen beskrivs med  $2 x$  där  $x$  är numret på den nod från vilken en boll ska plockas bort. Det garanteras att alla operationer är korrekta, d.v.s. det kommer inte krävas att lägga till fler bollar än det finns tomma noder eller att plocka bort en boll från en tom nod.

## Utdata

För varje operation av typ 1, skriv ut numret på den nod där den sist tillagda bollen hamnade. För varje operation av typ 2 skriv ut antalet bollar som rullade ner efter att bollen togs bort från den specificerade noden.

## Gränser

Alltid gäller  $N, Q \leq 100\,000$ .

I testfall värda 25 poäng har varje nod antingen 0 eller 2 barn. Dessutom ligger alla noderna med 0 barn lika långt från roten.

I testfall värda 30 poäng kommer operationerna i en sådan ordning att inga bollar någonsin kommer att rulla ner efter en operation av typ 2.

I testfall värda 40 poäng finns det exakt en operation av typ 1, och det är den allra första operationen.

Dessa tre delmängder av testfall är parvis disjunkta.

## Exempel

Indata	Utdata
8 4	1
0	3
1	2
2	2
2	
3	
3	
4	
6	
1 8	
2 5	
2 7	
2 8	

## Gränser

**Tidsgräns:** 1 sec per testfall

**Minnesgräns:** 128 MB per testfall