

Obligatorisk aflevering 2

Dette er den anden af tre afleveringsopgaver. Den skal løses individuelt, men må gerne diskuteres med de andre deltagere i kurset.

Problemformulering

Dit yndlingsskiresort ligger på et stort bjerg. Der går mange forskellige pister ned ad bjerget, med mange forgreninger, hvor en piste deler sig i flere. Det sker dog (besynderligt nok) aldrig at to pister mødes igen. En konsekvens af dette er at der, for alle steder på bjerget (pånær toppen), er præcis én måde at komme derned, omend der kan være flere måder at fortsætte længere ned. Fra bunden af alle pister, hvor man ikke kan komme længere, er der skilifter tilbage op til toppen af bjerget.

Som fast besøgende er du blevet bedt om at lave back-enden til en app der kan hjælpe med navigationen rundt på bjerget. Helt præcis skal du kunne svare på to slags forespørgsler: hvordan man hurtigst kommer fra a til b , og hvor langt der er fra a til b , hvor a og b kan være hvor som helst på bjerget.

Input

Første linje af input er et heltal, n , som angiver størrelsen af skibakken.

Næste linje indeholder $n - 1$ heltal, $0 \leq p_i < i$, som for alle $0 < i < n$ angiver det sidste sted man kommer til, inden man kommer til den i 'te position på skibakken. I graf termer er der en rettet kant fra p_i til i , som tager 1 minut at komme ned ad på ski. Toppen af skibakken er altid position 0. Hvis man ikke kan komme videre fra en position (udgrad er 0), så er der altid en skilift, der kan få dig tilbage til toppen af skibakken, og det tager altid 5 minutter.

Dernæst et der et heltal, q , som angiver antallet af forespørgsler omkring skibakken.

Til sidst følger q linjer med forespørgsler, som hver har en af de to former:

- **vej a b**, som beder om at få udskrevet den korteste vej fra a til b , hvor a aldrig er det samme som b .
- **tid a b**, som spørger om hvor lang tid det tager i minutter ad den korteste vej er fra a til b . Det tager 1 minut at stå på ski ad et piste-segment, og 5 minutter at tage en skilift op til toppen.

Output

q linjer med svar på forespørgslerne. **vej** forespørgsler besvares på formen $a \rightarrow i \rightarrow j \rightarrow k \rightarrow b$, mens **tid** besvares med et heltal.

Opgavedele

1. Skitser algoritmer, der løser problemerne, hvor du på abstrakt niveau henviser til de algoritmiske byggesten du anvender i din løsning. Det skal IKKE være en punktvis gennemgang af koden. Hvis du skriver “array” eller “variabel”, så er du gået for meget i detalje med en konkret implementation.
2. Skriv Java kode, der konkretiserer din beskrivelse ovenfor.
3. Analysér udførselstiden af programmet som funktion af n . **vej** og **tid** skal analyseres separat. Hvis du finder et tradeoff i udførselstid op mod at pre-beregne, så skal det også beskrives.

OBS

Det er ikke noget krav om at du finder den mest effektive løsning. Det vigtigste er at dine tre delopgaver passer sammen:

- Delopgave 1 skal være en kort og præcis overlevering af hvordan problemet kan løses. Tænk på det som det hint, du ville give en
- Javakoden skal implementere den algoritmiske idé du har beskrevet i delopgave 1.
- Køretidsanalysen skal være af Javakoden; ikke af hvad det bedst mulige kunne være.

Eksempel 1

input

```
7
0 1 1 3 4 3
6
vej 3 2
tid 1 3
vej 2 4
tid 2 6
vej 3 5
tid 0 5
```

output

```
3->6->0->1->2
1
2->0->1->3->4
8
3->4->5
4
```