

Software-installatie



Op een HPC (High Performance Computer) systeem worden zelden binaire pakketten geïnstalleerd, maar verkiest men doorgaans om de software van source te compileren. Op die manier is het mogelijk om de code platformafhankelijk te optimaliseren, wat doorgaans resulteert in betere prestatie. Dit betekent dat niet enkel de pakketten die onderzoekers uiteindelijk gebruiken gecompileerd zullen worden, maar ook alle afhankelijkheden of vereiste pakketten tot en met de systeembibliotheken. Op die manier bekomen we voor elk pakket p een zogenaamde afhankelijkheidsgraaf die aangeeft wat er moet beschikbaar zijn op het systeem alvorens het pakket p (dat zich ook in de graaf bevindt) op het systeem kan worden geïnstalleerd. Hieruit volgt meteen dat de volgorde waarin pakketten worden geïnstalleerd belangrijk is. Omdat het compileren en installeren van software gewoonlijk een tijdrovend proces is¹, kan het dus wel een tijd duren alvorens een pakket dat een onderzoeker vereist, beschikbaar is op het systeem. Het is echter vaak mogelijk om meerdere pakketten die niet afhankelijk zijn van elkaar op hetzelfde moment te compileren en te installeren. Hiervoor gebruiken we typisch meerdere machines in de cluster. Merk op dat we in wat volgt enkel spreken over installeren om de tekst wat beknopter te houden.

Opgave

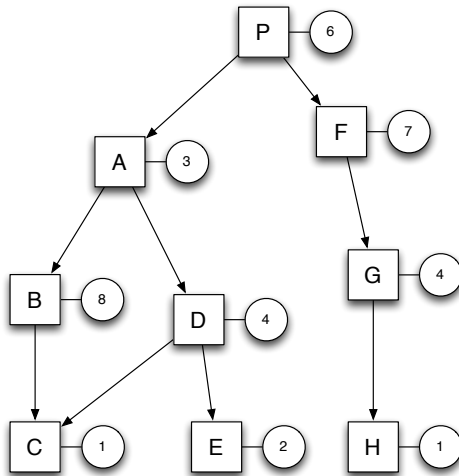
Je beschikt over de volgende gegevens:

- het aantal softwarepakketten,
- de tijd die nodig is om elk pakket te installeren,
- de afhankelijkheden tussen de pakketten,
- een gegeven vast aantal machines die tijdens de installatie kunnen aangewend worden.

Bepaal met deze gegevens hoe lang de installatieprocedure zal duren met het gegeven aantal machines dat ter beschikking is. Je zult hiervoor een installatievolgorde moeten bepalen, zodat de afhankelijkheden netjes worden gerespecteerd. Opdat de oplossing eenduidig zou zijn, moet je je houden aan de volgende randvoorwaarden.

1. Een pakket P is beschikbaar voor installatie op een tijdstip t als alle pakketten waarvan P afhankelijk is, geïnstalleerd zijn en er een machine vrij is op tijdstip t .

¹<http://xkcd.com/303>



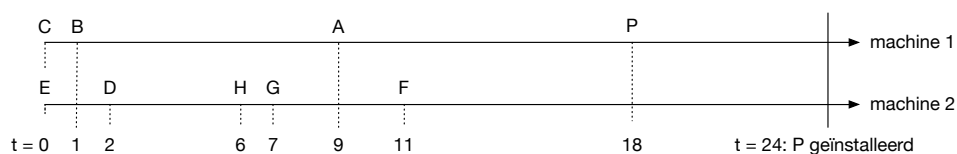
Figuur 1: Voorbeeld van een afhankelijkheidsgraaf om pakket P te installeren. De getallen geven de duur aan die nodig is om het corresponderende pakket te installeren. Een pijl van A naar B geeft aan dat B een afhankelijkheid vormt voor A en dat B dus voor A moet worden geïnstalleerd.

2. De installatie van pakket P kan bovendien enkel aanvangen op tijdstip t indien er geen ander pakket M bestaat dat alfabetisch voor P komt en dat ook beschikbaar is voor installatie op tijdstip t , m.a.w., als je uit meerdere pakketten kunt kiezen op tijdstip t , kies dan het pakket dat het eerst komt in de alfabetische rangschikking.

Merk op dat het bekomen resultaat *niet* noodzakelijk optimaal is wat de duur betreft, m.a.w., voor een andere volgorde is het best mogelijk dat het volledige installatieproces minder lang duurt. We vragen echter *niet* om te bepalen wat de optimale tijd is.

In het voorbeeld uit Figuur 1 krijgen we dan de installatievolgorde zoals getoond in Figuur 2. De eerste pakketten die in aanmerking komen voor installatie zijn C, E en H. Hiervan wordt C gekozen op machine 1. Op hetzelfde ogenblik kan machine 2 beginnen met de installatie van pakket E. Na 1 tijdseenheid is pakket C geïnstalleerd en wordt pakket B beschikbaar en wordt door machine 1 geïnstalleerd. Eenmaal pakket E geïnstalleerd is, wordt ook pakket D beschikbaar en wordt dit pakket gekozen vóór pakket H. Uiteindelijk zijn alle pakketten geïnstalleerd na 24 tijdseenheden.

In het bovenstaande voorbeeld was er slechts 1 top-level pakket, namelijk P, maar het is best mogelijk dat er meerdere dergelijke pakketten voorkomen. Het is wel degelijk de bedoeling dat alle pakketten uit de gegeven lijst geïnstalleerd worden.



Figuur 2: Voorbeeld van de installatievolgorde op de verschillende machines voor de pakketten en hun afhankelijkheden uit Figuur 1. Merk op dat van $t = 12$ tot $t = 18$ machine 1 ongebruikt blijft.

Invoer

De eerste regel van de invoer bevat een geheel getal $1 \leq n \leq 1000$ dat het aantal testgevallen aangeeft. Per geval volgen dan een aantal regels. Alle gegevens in de invoer die op dezelfde regel voorkomen, worden gescheiden door 1 enkele spatie; alle regels worden beëindigd met een enkele newline $\backslash n$.

Elk geval bestaat uit een aantal regels met informatie. De eerste regel geeft het aantal pakketten k aan. De tweede regel geeft aan hoeveel machines tijdens de installatieprocedure gebruikt kunnen worden. Daarna volgen k regels met op elke regel de naam van het pakket (bestaande uit 1 enkele hoofdletter), de tijd die nodig is om dat pakket te compileren en te installeren (uitgedrukt in minuten), gevolgd door de namen van de pakketten die afhankelijkheden vormen. Al deze gegevens worden telkens gescheiden door één enkele spatie. Zoals eerder vermeld, betekent dit dat het pakket P waarover informatie wordt verstrekt pas kan verwerkt worden nadat *alle* pakketten waarvan P afhankelijk is, zijn geïnstalleerd.

Uitvoer

De uitvoer bestaat uit n regels die voor elk geval twee gehele getallen afdrucken op 1 enkele regel, gescheiden door één enkele spatie. Het eerste getal duidt het volgnummer van het testgeval aan, het tweede getal de tijd die nodig is om voor dit testgeval de gegeven pakketten te installeren met het gegeven aantal machines.

Let op! Zorg ervoor dat je uitvoer geen overbodige tekens bevat, bijvoorbeeld een spatie op het einde van een regel of een lege regel op het einde van de uitvoer. Dat zorgt er immers voor dat je uitvoer als foutief wordt beschouwd.

Voorbeeld

Voor het voorbeeld uit Figuur 1, ziet de invoer er als volgt uit.

Invoer

1

9

2

A 3 B D

B 8 C

C 1

D 4 C E

E 2

F 7 G

G 4 H

H 1

P 6 A F

Uitvoer

1 24