

Rettelse til Opgave 2, jvf. »Opgave til stoffet gennemgået ved forelæsningen 4/11-2002«

Der var en fejl i den abstrakte algoritme. Her følger et forslag til en bedre; læg mærke til, at variablene har fået mere sigende navne og at der en mindre end før.

Forklaring på algoritmens variable:

næsteOpgaveAnkommer: Det tidspunkt ude i fremtiden, hvor det er forudset, at næste kunde kommer og afleverer en ny opgave. (Hvis værdien er kommet op på 480 eller derover betyder det, at denne nye opgave ikke kommer indenfor døren.)

næsteOpgaveKanPåbegyndes: Det tidspunkt ude i fremtiden, hvor A.Jensen kan påbegynde en ny opgave. (Men der skal selvfølgelig være en opgave, for at han kan gå løs på den.)

næsteOpgaveKanPåbegyndes = 0;
næsteOpgaveAnkommer = 0; // Dvs. første kunde antages at komme kl. 0

Gentag følgende:

```
Hvis næsteOpgaveAnkommer ≤ næsteOpgaveKanPåbegyndes
    og næsteOpgaveAnkommer < 480, så
{ Generér en ny opgave med tilfældig varighed mellem 1 og 30 minutter;
  Sæt opgave i kø eller stak som beskrevet;
  Generér et tidsinterval med tilfældig varighed D mellem 2 og 35 minutter
  for hvor længe der er til næste opgave ankommer;
  Sæt næsteOpgaveAnkommer = næsteOpgaveAnkommer + D;
}
Ellers // dvs. næsteOpgaveKanPåbegyndes < næsteOpgaveAnkommer
  Hvis der er en opgave at vælge,
    Tag en opgave ud af kø eller stak som beskrevet, og kald dens varighed V;
    Sæt næsteOpgaveKanPåbegyndes = næsteOpgaveKanPåbegyndes + V;
  Ellers // Der er ingen opgave som venter
    Hvis næsteOpgaveAnkommer ≥ 480
      // Fyraften og alle opgaver, som skal løses, er løst
      Stop
    Ellers
      // Der ingen opgaver nu, men vi ved der ankommer en om lidt
      næsteOpgaveKanPåbegyndes = næsteOpgaveAnkommer
```