488 (sieve) Given variable p: [n*bin] := [⊥; ⊥; (n–2)*⊤] , the following program is the sieve of Eratosthenes for determining if a number is prime.

for i:= 2;..$ceil$ ($n^{1/2}$)
do if p i **then for** j:= i;..$ceil$ (n/i) **do** p:= (i×j)→⊥ | p **od**
else ok **fi od**

(a) Show how the program can be transformed for concurrency.

(b) What is the execution time, as a function of n , with maximum concurrency?

After trying the question, scroll down to the solution.

(a) Show how the program can be transformed for concurrency.

§ For any particular values of n and i, the values of $i \times j$ for j: $i,..ceil\ (n/i)$ are all different, so the iterations of the inner **for**-loop

$$\textbf{for } j := i;..ceil\ (n/i) \textbf{ do } p := (j \times i) \to \perp \mid p \textbf{ od}$$

can be executed concurrently. The iterations of the outer **for**-loop cannot be executed concurrently.

(b) What is the execution time, as a function of n, with maximum concurrency?

§ Putting $t := t+1$ inside the outer loop only,

$$\textbf{for } i := 2;..ceil\ (n^{1/2})$$
$$\textbf{do if } p\ i \textbf{ then for } j := i;..ceil\ (n/i) \textbf{ do } p := (i \times j) \to \perp \mid p \textbf{ od}$$
$$\textbf{else } ok \textbf{ fi}.\ \ t := t+1 \textbf{ od}$$

the execution satisfies

$$t' = t + ceil\ (n^{1/2}) - 2$$