

Your answers must be in English, presented as a text file (no other formats are accepted), and submitted using the web interface. Write `Omega` for Ω , `0` for O , `Theta` for Θ , `n^a` for n^a , and `sqrt(n)` for \sqrt{n} .

A Listfunktioner

Förklara Θ -uttryck för följande listfunktioner i SML:

1. `null xs`
2. `tl xs`
3. `List.last xs`
4. `length xs`
5. `xs @ ys`

var `xs` och `ys` är listor med n respektive m element.

B Master Theorem

Lös följande rekursionsformler med hjälp av *Master Theorem* (MT). Motivera genom att ange vilket fall du använder. Vissa (exakt 1) går inte att lösa med MT utan man måste tillämpa andra metoder: ange på dessa varför inte MT är tillämpbar.

1. $T(n) = 3T(\frac{n}{2}) + n^2$
2. $T(n) = 4T(\frac{n}{2}) + n^2$
3. $T(n) = 5T(\frac{n}{2}) + n^2 \lg n$
4. $T(n) = T(n-1) + n$
5. $T(n) = 8T(\frac{n}{3}) + 2^n$
6. $T(n) = 2T(\frac{n}{4}) + c$
7. $T(n) = 2T(\frac{n}{4}) + \sqrt{n}$
8. $T(n) = 4T(\frac{n}{2}) + \lg n$

C Analysera

Betrakta följande program i SML:

```
fun rev [] = []
  | rev (e :: es) = rev es @ [e]
```

1. Ge en rekursionsformel för körtiden för `rev xs` om listan `xs` har längd n . Anta att basfallet tar c_0 tidsenheter, att konstruera en lista med ett element tar c_1 tidsenheter, och att ett anrop till `ys @ zs` tar $c_2 * \text{length}(ys)$ tidsenheter.
2. Expandera rekursionsformeln 5 gånger, dvs ge värden för $T(0), T(1), \dots, T(4)$.
3. Visa att rekursionsformeln är ekvivalent med $c_0 + nc_1 + \frac{n(n-1)}{2}c_2$. (Tips: Prova med ett induktionsbevis.)
4. Visa att $c_0 + nc_1 + \frac{n(n-1)}{2}c_2 = \Theta(n^2)$.