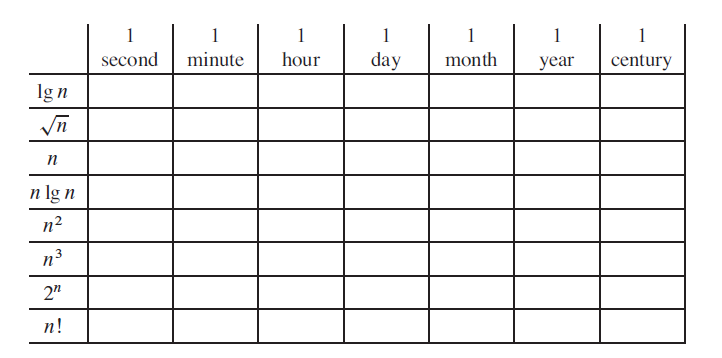
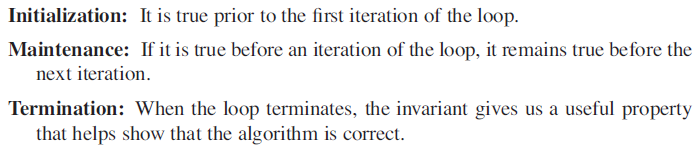
**Order of growth**



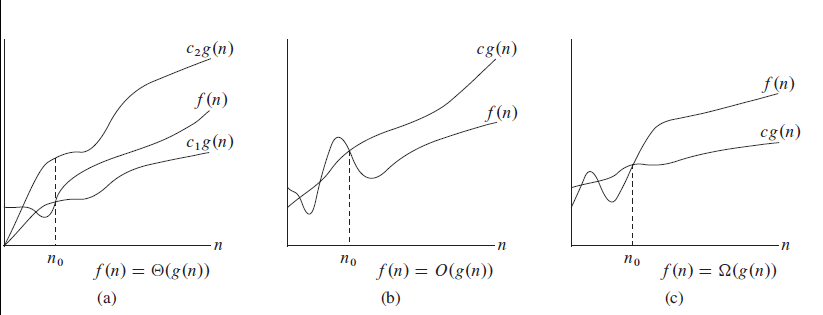
Order of growth er baseret på det største led i køretid. F.eks n^2 + n + lg(n) = theta(n). Dette er fordi de mindre led er ligegyldige for støre værdier af n.

**Looping invariants**

Looping invariants bliver brugt til at forstå hvorfor et algoritme er korrekt. Der skal bevises 3 ting i en looping invariant:



**Asymtotic Notations**



Vi siger at fra værdien n0 og alle værdier af n (eller, for alle n, hvor n >= n0) skal der findes constanter som gør f(n) holder sig imellem c1 \* g(n) og c2 \* g(n). Dette betyder at der er et *asymtotically tight bound for f(n).* Θ-notationen kræver også at alle f(n) i Θ(g(n)) er non-negative (figur a).

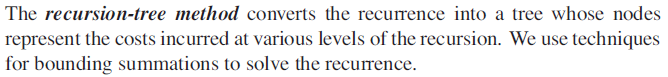
O(n) (Store O af n) giver et asymtotic upper bound. Dette vil sige at alle værdier i f(n) er mindre eller lig med O(g(n)). Fordi den altid laver et ”upper bound”. Omega notationen laver altid et ”lower bound” fordi alle værdier i f(n) er størrer end omega(g(n)). Vi bruger lille omega og o, til at sige der ikke er et asymtotic bound.

**The Substitution Method**

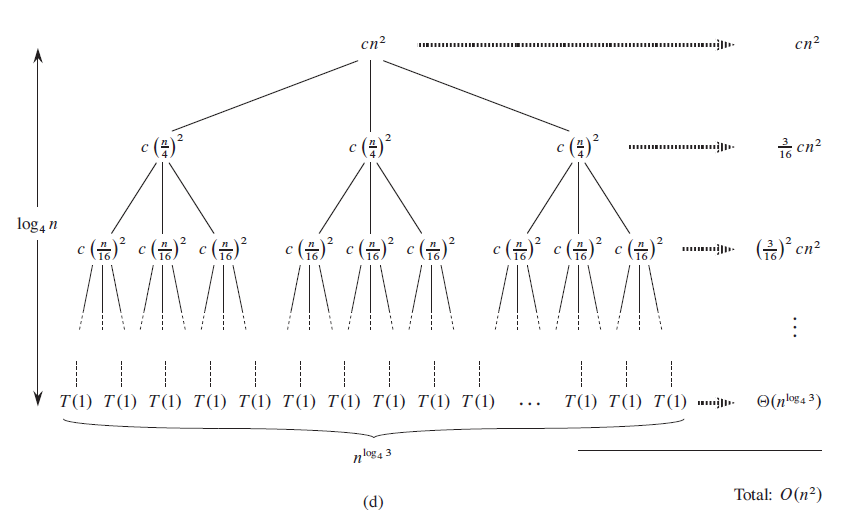


Denne metode handler om at lave et gæt på formen af løsningen (vil være inkluderet i opgaven) og derefter bruge matematisk induktion til at bevise løsningen. (se kapitel 4.3)

**The Recursion-Tree method**



Recursion-trees bliver brugt til at lave et gæt til substitutions metoden. Her tegner man et træ hvor noderne repræsentere et del-problem (hvor ”dyrt” er et del-problem at løse). Vi summere cost i vært niveau af træet og summere vært niveau for at få den totale værdi.



Figuren viser recurrence: T(n) = 3T (n/4) + cn^2

Denne metode er bedst brugt til at finde et gæt til at godkende med substitutions metoden.

**The Master Method**

Master metoden afhænger af følgende sætning:

