Volgende stelling uit 1965, van de hand van Juris Hartmanis en Richard Stearns, de uitvinders van computationele complexiteit, was een van de allereerste resultaten in de complexiteitstheorie.

Space Hierarchy Stelling. *Zij* $f : \mathbf{N} \to \mathbf{N}$ *zo dat* $f(n)$ *berekenbaar is uit* 1^n *in space* $O(f(n))$. *Zij* $g : \mathbf{N} \to \mathbf{N}$ *zo dat* $g \in o(f)$ *(d.w.z.,* $\lim_{n\to\infty} g(n)/f(n) = 0$*). Dan zijn er problemen die wel in* $\mathrm{SPACE}(f)$ *zitten maar niet in* $\mathrm{SPACE}(g)$*, of kort:*

$$\mathrm{SPACE}(g) \subset \mathrm{SPACE}(f)$$

waar '$\subset$*' staat voor* strikte *inclusie.*

Bewijs. Beschouw volgend programma D:

1. Input w. Zij n de lengte van w.
2. Baken $f(n)$ cellen af op de werktape. Dit kan binnen $f(n)$ space dankzij de onderstelling op f. Wanneer we in het vervolg van het programma ooit buiten deze afbakening zouden willen komen, wordt het programma afgebroken met een reject. We zorgen er dus voor dat ons programma onmogelijk meer dan $f(n)$ space kan gebruiken.
3. Test of de input w van de vorm $\langle M\rangle 0^*$ is, dus, een beschrijving van een Turing machine, eventueel gevolgd door een aantal nullen. Indien niet, reject.
4. Simuleer M op w. Indien w aanvaard wordt door M, rejecten we w. Indien w gereject wordt door M, aanvaarden we w. Omdat we de tapesymbolen en de toestanden van M voorstellen door getallen, b.v., in binair, gebruiken we $dg(n)$ space voor deze simulatie, waar d het aantal bits is die we nodig hebben om 1 tapesymbool of toestand van M voor te stellen.

Zij A nu de taal aanvaard door D (dus de verzameling van alle strings aanvaard door D). Dan zit A zeker in $\mathrm{SPACE}(f)$, we hebben D immers zo gemaakt dat het altijd binnen space $f(n)$ blijft. We gaan nu argumenteren dat A niet in $\mathrm{SPACE}(g)$ zit voor elke $g \in o(f)$.

Beschouw daartoe een willekeurige Turing machine M die binnen space $g(n)$ werkt, met $g \in o(f)$. Wanneer we D runnen op een input $w = \langle M\rangle 0^*$, zal D de simulatie van M op w niet voortijdig afbreken op voorwaarde dat $dg(n) \leqslant f(n)$. Immers, $dg(n)$ is de space nodig om de simulatie uit te voeren, zoals hierboven opgemerkt. Omdat $g \in o(f)$ kunnen we gelukkig het aantal nulletjes groot genoeg kiezen zodat inderdaad $dg(n) \leqslant f(n)$. Immers, omdat $\lim_{n\to\infty} g(n)/f(n) = 0$ is er voor elke $\varepsilon > 0$ een n_0 zodat $g(n)/f(n) < \varepsilon$ voor elke $n \geqslant n_0$. Neem dan $\varepsilon = 1/d$ en we krijgen $dg(n) < f(n)$ voor elke $n \geqslant n_0$. Als we dus input $w = \langle M\rangle 0^{n_0}$ nemen dan is de lengte $n \geqslant n_0$ en is dus $dg(n) \leqslant f(n)$. De simulatie van M op input w door D wordt dus niet afgebroken. Als w aanvaard wordt door M, zal w gereject worden door D, en dus niet in A zitten. Als w gereject wordt door M, zal w aanvaard worden door D, en dus wel in A zitten. De taal aanvaard door M is dus *niet* gelijk aan A.

We concluderen dat A verschilt van elke taal in SPACE(g) voor $g \in o(f)$, zoals gewenst. □

Enkele gevolgen:

1. SPACE(n) $\subset$ SPACE(n^2) $\subset$ SPACE(n^3) $\subset \cdots$
2. L $\subset$ PSPACE
3. PSPACE $\subset$ EXPSPACE
4. Geen enkel EXPSPACE-compleet probleem zit in PSPACE, en dus zeker niet in P. Inderdaad, anders zou heel EXPSPACE in PSPACE zitten (waarom?) en we hebben net gezien dat dit niet waar is.

   Er zijn dus problemen die beslisbaar zijn, maar waarvoor bewijsbaar geen efficiënt algoritme bestaat. Een concreet voorbeeld van een EXPSPACE-compleet probleem is het volgende: (waarom?)

$$U = \{\langle M, w, 1^s\rangle \mid M \text{ aanvaardt } w \text{ in space } 2^s\}$$

We besluiten met op te merken dat er een analoge, maar ietsje minder nauwkeurige Hierarchy Stelling voor tijd i.p.v. space is:

Time Hierarchy Stelling. *Zij $f : \mathbf{N} \to \mathbf{N}$ zo dat $f(n)$ berekenbaar is uit 1^n in tijd $O(f(n))$. Zij $g : \mathbf{N} \to \mathbf{N}$ zo dat $g \log g \in o(f)$. Dan is* TIME(g) $\subset$ TIME(f).

De gevolgen zijn analoog: TIME(n) $\subset$ TIME(n^2) $\subset$ TIME(n^3) $\subset \cdots$, en ook P $\subset$ EXPTIME, en dus ook dat geen enkel EXPTIME-compleet probleem in P zit. Een concreet voorbeeld van een EXPTIME-compleet probleem is het volgende: (waarom?)

$$U = \{\langle M, w, 1^t\rangle \mid M \text{ aanvaardt } w \text{ in tijd } 2^t\}$$