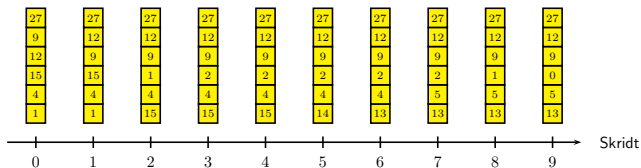


Invarianter

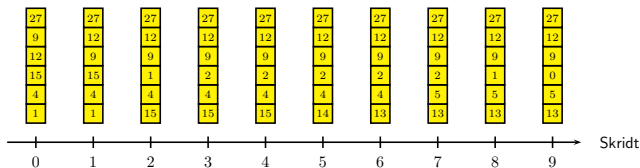
State

State for algoritme: dens nuværende indhold af hukommelsen (incl. registre). Hver basalt skridt i algoritme \Rightarrow mulig ny state:



State

State for algoritme: dens nuværende indhold af hukommelsen (incl. registre). Hver basalt skridt i algoritme \Rightarrow mulig ny state:



Husk definition af korrekthed af algoritmer:

- ▶ Terminerer (for alle input).
- ▶ Korrekt output (for alle input) når terminerer.

For at kunne vise ovenstående må vi vide noget om hvordan state udvikler sig. Fanges ofte ved hjælp af **invarianter** og **termineringsfunktioner**.

Invarianter

Invariant for algoritme: et udsagn $S(k)$ om state som:

- ▶ Er sandt efter skridt k for alle skridt k .
- ▶ Ved algoritmens afslutning kan korrekthed af output udledes af udsagnet $S(k)$ samt de omstændigheder som fik algoritmen til at stoppe.

I praksis ofte een invariant for een samling skridt (f.eks. 0 eller flere gennemløb af en **while** eller **for** løkke). En anden invariant kan så om nødvendigt bruges for næste samling skridt af algoritmen.

Termineringsfunktion

En (ofte meget simpel) funktion af state som

- ▶ Altid er ≥ 0 .
- ▶ Falder med mindst 1 for hvert skridt.

Hvis en funktion med ovenstående egenskaber kan defineres, er terminering klar: der kan kun foretages et antal skridt svarende til funktionens værdi ved start af algoritmen (dvs. dens værdi for state til tiden nul).

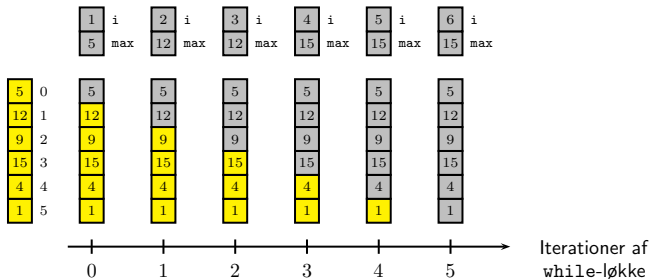
Eksempel

Find største element i array:

```
max = A[0]
i = 1
while i < A.length
  if A[i] > max
    max = A[i]
  i++
```

Invariant $S(k)$: "Efter den k 'te iteration af while-løkke indeholder max den største værdi af $A[0..(i-1)]$ ".

Termineringsfunktion: $A.length - i$.



Induktion

Bemærk at dette er et tilfælde af induktion

$$\begin{array}{l} 1) S(1) \text{ er sand} \\ 2) S(k) \text{ sand} \Rightarrow S(k+1) \text{ sand} \end{array} \Rightarrow S(k) \text{ sand for alle } k$$

hvor k er antal iterationer af løkken.

Brug af invarianter

Invarianter kan bruges på forskellige detalje-niveauer (med en glidende overgang imellem dem):

- ▶ Som værktøj til at udvikle algoritme-ideer: Med den rette invariant fanges essensen af metoden, og algoritmen skal “blot” skrives ud fra at denne invariant skal vedligeholdes.
- ▶ Som værktøj til at nedskrive konkret kode (eller detaljeret pseudo-kode) og vise den korrekt.

I første anvendelse er figurmæssige betragtninger passende. I anden anvendelse må man nedskrive invarianten i termer af konkrete variable og andre dele af state, samt argumentere om disses opdatering af den konkrete kode.

Eksemplet ovenfor med af finde største element i et array illustrerer begge dele (figur samt invariant nedskrevet som $S(k)$). Yderligere eksempler i bogen er f.eks.: PARTITION (afsnit 7.1), INSERTION-SORT (afsnit 2.1) og MERGE (afsnit 2.3.1). Mange flere vil blive mødt senere i kurset.