

FÖRELÄSNING 12

ALGORITMKONSTRUKTION 3

DYNAMISK PROGRAMMERING

1. BESTÄM STRUCTUREN HOS OPTIMAL LÖSNING
2. STÄLL UPP REKURSION FÖR OPTIMALVÄRDET
3. BERÄKNA DELPROBLEMENS OPTIMALVÄRDEN FRÅN SMÅ TILL STORA
4. KONSTRUERA OPTIMALLÖSNINGEN (EV. MED EXTRA INFORMATION SOM LAGRAS I STEG 3)

1. OPTIMALA LÖSNINGENS STRUKTUR

STIG UPPBYGGD AV DELSTIGAR

2. REKURSION

LÅT $V[i,j]$ = VÄRDET PÅ BÄSTA STIGEN FRÅN $a_{i,j}$ NER TILL RAD n .

$$V[i,j] = \begin{cases} a_{i,j} & \text{OM } i=n \text{ (DVS SISTA RADEN)} \\ a_{i,j} + \max(V[i+1,j], V[i+1,j+1]) & \text{ANNARS} \end{cases}$$

3. BERÄKNING

FOR $j \leftarrow 1$ TO n DO
 $V[n,j] \leftarrow a_{n,j}$

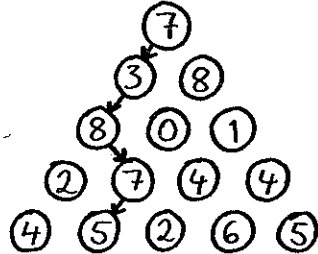
FOR $i \leftarrow n-1$ DOWNTO 1 DO
 FOR $j \leftarrow 1$ TO i DO
 $V[i,j] \leftarrow a_{i,j} + \max(V[i+1,j], V[i+1,j+1])$

RETURN $V[1,1]$

TIDSKOMPLEXITET: $O(n^2)$

EXEMPEL 1: MAXIMAL TRIANGELSTIG

PROBLEM: HITTA STIGEN FRÅN TOPPEN NER TILL BOTTEN SOM MAXIMERAR SUMMAN AV DOM INGÅENDE TALEN.



n = ANTAL RADER

FINNS 2^{n-1} STIGAR ATT PRÖVA

$a_{i,j}$ = ELEMENT NR j PÅ RAD i

Exempel på dynamisk programmering

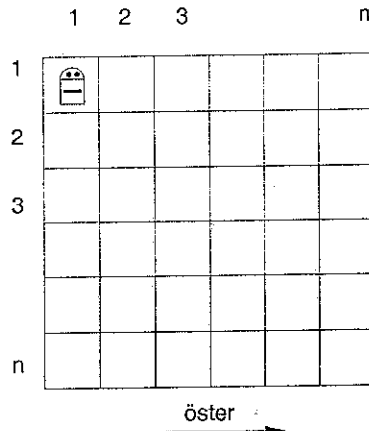
Roboten R2D2 ska röra sig i ett rutnät med n^2 rutor. R2D2 startar i ruta $(1,1)$ och ska ta sig till ruta (n,n) . Han rör sig med ett steg per sekund, antingen österut eller söderut. Har han otur tar batteriet slut och han blir stående. Om han går öster eller söder om rutnätet ramlar han av och är förlorad.

I varje ruta (i,j) finns två tal $p_{i,j}$ och $q_{i,j}$. När R2D2 står på ruta (i,j) gäller följande:

R2D2 går åt öster med sannolikhet $p_{i,j}$

han går åt söder med sannolikhet $q_{i,j}$

batteriet tar slut med sannolikhet $1 - (p_{i,j} + q_{i,j})$.



Beskriv sannolikheten för att R2D2 når rutan (i,j) med en rekursionsekvation och använd den för att konstruera en algoritm som räknar ut sannolikheten att roboten ska nå fram till rutan (n,n) .