

Uppgifter till föreläsning 2 om dynamisk programmering, DD2350 adk20

1. Matriskedjemultiplikation

Matriskedjemultiplikation är problemet att hitta bästa sättet att multiplicera ihop en kedja av matriser av varierande storlek. Genom att multiplicera ihop matriserna med varandra i en listig ordning kan antalet operationer som krävs minimeras. Att multiplicera en $a \times b$ -matris och en $b \times c$ -matris med vanlig matrismultiplikation tar abc talmultiplikationer.

Vi vill beräkna matrisprodukten $A_1 A_2 \cdots A_n$ där matris A_i har dimension $r_{i-1} \times r_i$.

Låt $M[i, j]$ = minimala antalet multiplikationer som behövs för att räkna ut matrisprodukten $A_i A_{i+1} \cdots A_j$.

Lista ut vad ??? ska vara i följande rekursion för $M[i, j]$.

$$M[i, j] = \begin{cases} 0 & \text{om } i = j \\ \min_{i \leq k < j} (???) & \text{om } i < j \end{cases}$$

Ledning: k står för den position i matriskedjan där den sista matrismultiplikationen görs när produkten $A_i A_{i+1} \cdots A_j$ ska beräknas.

2. Robotproblemet

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.

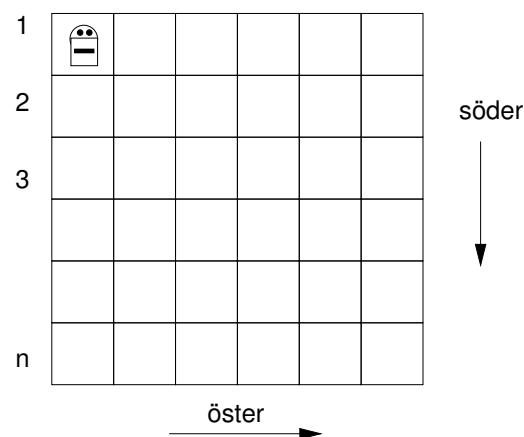
I varje ruta (i, j) finns två tal $p_{i,j}$ och $q_{i,j}$. När R2D2 står på ruta (i, j) gäller följande:

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

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

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

1 2 3 n



Beskriv sannolikheten för att R2D2 når rutan (i, j) med en rekursionsekvation och ta fram en beräkningsordning som kan användas för en dynamiskprogrammeringsalgorithm som räknar ut sannolikheten att roboten ska nå fram till rutan (n, n) .

3. Hållbart fiske

Experter på hållbar utveckling har utformat ett ganska komplicerat kvotsystem för att begränsa fiske, ett system som bygger på *fiskebiljetter*. Varje fiskare får k stycken fiskebiljetter numrerade 1 till k och som ska användas i nummerordning. Varje gång fiskaren fiskar gör han eller hon av med en fiskebiljett. Hur mycket fisk som fiskaren får dra upp då beror på för vilken gång i ordningen fiskaren fiskar, alltså fiskebiljettens nummer. Man får bara fiska en gång vid varje fiskeplats.

Anta att en fiskare åker en rutt som i tur och ordning passerar n stycken lämpliga fiskeplatser, numrerade 1 och n , och fiskaren vill fiska vid k av dessa fiskeplatser. Fiskaren vill veta vilka k platser hon ska fiska vid för att få mest fisk.

Din uppgift är att konstruera dynamiskprogrammeringsdelproblem, en rekursion för delproblemen och svaret uttryckt i delproblemen, för problemet att givet fiskekvoterna $K_{i,j}$ (som står för hur många fiskar man får ta upp vid fiskeplats i om man använder fiskebiljett j), där $1 \leq i \leq n$ och $1 \leq j \leq k$, beräkna den maximala mängden fisk som kan fångas om man väljer dom k fiskeplatserna optimalt. Alla fiskebiljetterna behöver inte gå åt.

Lösningar

1. Matriskedjemultiplikation

Minimala antalet multiplikationer som behövs för att räkna ut matrisen $A_i \cdots A_k$ är $M[i, k]$ stycken. Denna matris har dimensionen $r_{i-1} \times r_k$. Minimala antalet multiplikationer som behövs för att räkna ut matrisen $A_{k+1} \cdots A_j$ är $M[k+1, j]$ stycken. Denna matris har dimensionen $r_k \times r_j$. Att multiplicera en $r_{i-1} \times r_k$ -matris med en $r_k \times r_j$ -matris tar $r_{i-1}r_kr_j$ talmultiplikationer.

Rekursionsekvationen för $M[i, j]$ blir därför:

$$M[i, j] = \begin{cases} 0 & \text{om } i = j \\ \min_{i \leq k < j} (M[i, k] + M[k+1, j] + r_{i-1}r_kr_j) & \text{om } i < j \end{cases}$$

Förra gången utformade vi en beräkningsordning och en algoritm som beräknar M .

2. Robotproblemet

Låt sannolikheten för att R2D2 når rutan (i, j) vara $P[i, j]$.

Basfallet är $P[1, 1] = 1$.

Generella fallet är $P[i, j] = P[i, j - 1] \cdot p_{i,j-1} + P[i - 1, j] \cdot q_{i-1,j}$.

För första raden och första kolumnen i matrisen kommer bara en av termerna i rekursionen att finnas.

Den sökta sannolikheten är $P[n, n]$.

Flera beräkningsordningar är tänkbara, till exempel radvis uppifrån och ner och inom varje rad från vänster till höger. Kolumnvis går också bra.

3. Hållbart fiske

Låt $F[i, j]$ vara maximala antalet fiskar som kan fås vid fiskeplats $1 \dots i$ med användning av dom j första biljetterna för alla $1 \leq i \leq n$ och $0 \leq j \leq k$.

Om det hade krävts att alla k fiskebiljetterna ska användas skulle vi hitta lösningen i $F[n, k]$. Eftersom det är tillåtet att inte använda dom sista fiskebiljetterna så är den maximala lösningen istället $\max_{1 \leq j \leq k} F[n, j]$.

Rekursionen blir

$$F[i, j] = \begin{cases} 0 & \text{om } j = 0 \\ K_{1,1} & \text{om } i = j = 1 \\ -\infty & \text{om } i = 1, j > 1 \\ \max(F[i-1, j], F[i-1, j-1] + K_{i,j}) & \text{om } i > 1, j > 0 \end{cases}$$