Question 1: Grunnlegende

Hva vil det si at et problem har optimal substruktur?

- ( ) At problemet kan løses i polynomisk tid
- ( ) At delinstansene overlapper
- (•) At dersom vi løser delinstansene optimalt kan vi også løse problemet optimalt
- ( ) At optimalisering av strukturen kan gjøres i konstant tid

Question 2: Grunnlegende

Hva innebærer overlappende delinstanser?

- ( ) At delinstansene er praktisk talt like, og derfor trygt kan ignoreres
- (•) At samme delinstansen må løses flere ganger av en rekursiv algoritme
- ( ) At delinstansene blir løst i konstant tid
- ( ) Ingen av delene

Question 3: Grunnlegende

Hvilket av disse problemene er hensiktsmessig å løse med dynamisk programmering?

- ( ) Sortere en liste bestående av n heltall mellom 1 og n.
- (•) Finne det n'te Fibonacci tallet
- ( ) Finne det elementet i et binært søketre som har verdi nærmest en oppgitt verdi.
- ( ) Finne et element som forekommer mer enn én gang i en liste bestående av n heltall mellom 1 og n-1.

Question 4: Rekursive problemer

Merge-sort er et eksempel på en algoritme som rekursivt løser et problem. Hvorfor kan vi ikke bruke dynamisk programmering til å forbedre den?

- ( ) Fordi problemet ikke har optimal subtruktur
- ( ) Fordi sortering ikke tar eksponensiell tid
- ( ) Fordi merge-sort ikke er in-place.
- (•) Fordi delproblemene ikke overlapper

Question 5: Rekursive problemer

Alle riktige svar funnet.

Ett rett svar.

Ett rett svar.

Hvilke(n) av disse rekursive dekomponeringene beskriver et problem som trolig kan løses ved hjelp av dynamisk programmering?

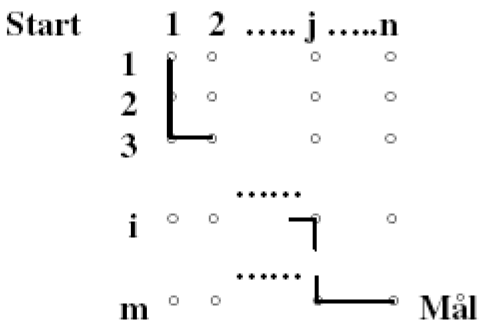
Hint: Prøv å tegne delinstans-grafen og se etter overlapp, start med f.eks. P(5,5) og se om noen av delinstansene blir like.

Hva basis-tilfellene for rekkurensene er ikke viktig i denne oppgaven, bare anta at de finnes.

- [x] $P(i,j) = \max\{P(i-1,j), P(i,j-1), P(i-1,j-1)\}$
- [x] $P(i,j) = \min\{P(i,j-2), P(i-1,j)\}$
- [ ] $P(i,j) = P(i-1,j-1) + 2$
- [ ] $P(i,j) = 4$

Question 6: Matrisetraversering

I denne oppgaven skal vi ta for oss et rektangulært rutenett gitt som følger:



Vi skal nå prøve å finne ut av hvor mange veier det finnes fra punkt start (punkt [1, 1]) til punkt Mål (punkt [m, n]) under visse restriksjoner. En lovlig vei fra Start til Mål defineres ved at et skritt fra punkt [i, j] på veien skal gå enten til punktet [i+1, j] eller til punktet [i, j+1]. To veier er forskjellige dersom de ikke er identisk like, skritt for skritt. Funksjonen T(i, j) skal gi antall veier fra punkt [1, 1] til punkt [i, j]. Dette fører til at T(1, 2) = 1 og T(3, 2) = 3.

Hva blir T(1,4)?

- ○ 5
- ○ 2
- ● 1
- ○ 4
- ○ 3

Question 7: Matrisetraversering

Hva blir T(6,3)? (Det kan være lurt å finne et system)

- ( ) 8
- ( ) 14
- ( ) 10
- (•) 21
- ( ) 18

Question 8: Matrisetraversering

I dynamisk programmering handler det ofte om å finne et utrykk som gir deg svaret på et problem dersom du allerede har svaret på en delinstans av problemet, en rekursiv dekomponering.

Hvilket av utrykkene under beskriver T(m,n)?

T(m,n) =

- ( ) m * 42
- ( ) T(m-1,n-1) + 2
- ( ) ı m^2 + n^2 - m * n
- (•) T(m-1, n) + T(m, n-1)
- ( ) MAX{ T(m-1, n), T(m, n-1) } + 1

Question 9: Stavkutting

Gitt en stav med lengde N. En stav med lengde i kan selges for p_i, for i=1,2,...,N.

Finn hvordan staven skal kuttes opp slik at du maksimerer inntekten R ved å selge staven.

Hva blir inntekten R når

N = 4

p_1 = 3, p_2 = 7, p_3 = 12, p_4 = 13

- (•) 15
- ( ) 17
- ( ) 13
- ( ) 14
- ( ) 16

Question 10: Stavkutting

Hva blir R når

N = 8

p_1 = 3, p_2 = 4, p_3 = 7 p_4 = 10, p_5 = 12, p_6 = 14, p_7 = 15, p_8 = 18

- ○ 23
- ○ 26
- ◉ 24
- ○ 18

Question 11: Stavkutting

Hvor mange delinstanser må man løse for å finne optimal løsning for stavkutte-problemet hvis staven har en lengde n?

Merk: Her er det ikke viktig hvor lang tid en algoritme ville brukt på å løse problemet.

- ◉ Θ(n)
- ○ Θ(n^2)
- ○ Θ(2n)
- ○ Θ(lg n)