

Algoritmer, datastrukturer och komplexitet, hösten 2016

Uppgifter till övning 4

Dynamisk programmering

På denna övning är det också **inlämning av skriftliga lösningar av teoriuppgifterna till labb 2** och muntlig redovisning av teoriuppgifterna.

Träskvandring Tina ska gå genom ett träsk som representeras av ett $n \times n$ -rutmönster från vänsterkanten till högerkanten. I varje steg kan hon gå ett steg rakt till höger, snett uppåt höger eller snett nedåt höger. Det är olika jobbigt att gå på olika ställen i träsket. Att hamna på ruta (i, j) kostar arbetet $A[i, j]$, som är ett positivt heltal.

Beskriv en algoritm som hittar det minimala arbetet att ta sig igenom träsket i följande olika fall.

- Anta att Tina får starta var som helst på vänsterkanten och får sluta var som helst på högerkanten.
- Anta att Tina alltid startar i ruta $(n/2, 1)$.
- Anta att Tina dessutom måste sluta i ruta $(n/2, n)$.
- Skriv ut den minst jobbiga väg Tina kan ta genom träsket i c).

Kombinera mynt och sedlar Givet en (konstant) uppsättning mynt- och sedelslag med värdena v_1, \dots, v_k kronor och ett maximalt belopp n . (Alla tal i indata är positiva heltal.) Formulera en rekursion för på hur många sätt man kan bilda varje summa mellan 0 och n kronor med hjälp av denna uppsättning mynt och sedlar.

Längsta gemensamma delsträng Strängarna ALGORITM och PLÅGORIS har den gemensamma delsträngen GORI. Den *längsta gemensamma delsträngen* hos dessa strängar har alltså längd 4. I en delsträng måste tecknen ligga i en sammanhängande följd.

Konstruera en effektiv algoritm som givet två strängar $a_1a_2 \dots a_m$ och $b_1b_2 \dots b_n$ beräknar och returnerar längden hos den längsta gemensamma delsträngen. Algoritmen ska bygga på dynamisk programmering och gå i tid $O(nm)$.

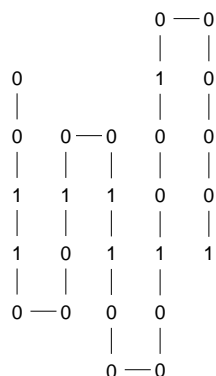
Här följer två roliga men mer komplicerade exempel som vi nog inte hinner med på övningen:

Proteinvikning Ett protein är en lång kedja av aminosyror. Proteinkedjan är inte rak som en pinne utan hopvikt på ett intrikat sätt som minimerar den potentiella energin. Man vill väldigt gärna kunna räkna ut hur ett protein kommer att vika sig. I denna uppgift ska vi därför studera en enkel modell av proteinvikning där aminosyrorna är antingen *hydrofoba* eller *hydrofila*. Hydrofoba aminosyror tenderar att klumpa ihop sig.

För enkelhets skull ser vi proteinet som en binär sträng där ettor motsvarar hydrofoba aminosyror och nollor hydrofila aminosyror. Strängen (proteinet) ska sedan vikas i ett tvådimensionellt kvadratisk gitter. Målet är att få dom hydrofoba aminosyrorna att klumpa ihop sig, det vill säga att få så många ettor som möjligt att ligga nära varandra. Vi har alltså ett optimeringsproblem där målfunktionen är antalet par av ettor som ligger intill varandra i gittret (lodrätt eller vågrätt) utan att vara intill varandra i strängen.

Du ska konstruera en algoritm som med hjälp av dynamisk programmering konstruerar en optimal *dragspelsvikning* av en given proteinsträng av längd n . En dragspelsvikning är en vikning där strängen först går en sträcka rakt nedåt, sedan en sträcka rakt uppåt, sedan en sträcka rakt nedåt, och så vidare. I en sådan vikning kan man notera att lodräta par av intilliggande ettor alltid kommer i följd i strängen, så det är bara vågräta par av ettor som bidrar till målfunktionen.

I följande figur är strängen 00110001001100001001000001 dragspelsvikt på ett sådant sätt att målfunktionen blir 4.



Definition av problemet PROTEINDRAGSPELSVIKNING:

INMATNING: En binär sträng med n tecken.

PROBLEM: Hitta den dragspelsvikning av indatasträngen som ger det största värdet på målfunktionen, alltså det största antalet par av ettor som ligger bredvid varandra men inte direkt efter varandra i strängen.

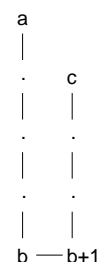
Konstruera och analysera tidskomplexiteten för en algoritm som löser proteindragspelsvikningsproblemet med dynamisk programmering.

Du får gärna använda dig av nedanstående algoritm, som beräknar antalet par ettor i ett varv (dvs mellan två sträckor) i en dragspelsvikning som ligger bredvid varandra (men inte direkt efter varandra i strängen). Anta att proteinet lagras i en array $p[1..n]$. Parametrarna a och b anger index i arrayen för den första sträckans ändpunkter. Parametern c anger index för den andra sträckans slutpunkt. Se figuren nedanför till höger.

```

profit(a,b,c) =
  shortest ← min(b-a, c-(b+1));
  s ← 0;
  for i ← 1 to shortest do
    if p[b-i]=1 and p[b+1+i]=1 then
      s ← s+1;
  return s;

```



Not: Proteinvikningsproblemet är ett viktigt algoritmiskt problem som studeras i bioinformatiken. Det behandlas tillsammans med många andra problem med biologisk anknytning i den valfria kursen *Algoritmisk bioinformatik* som går i period 4 varje år.

Analysator för kontextfri grammatik En *kontextfri grammatik* brukar användas för att beskriva syntax för bland annat programspråk. En kontextfri grammatik i *Chomskynormalform* beskrivs av

- en mängd slutsymboler T (som brukar skrivas med små bokstäver),
- en mängd ickeslutsymboler N (som brukar skrivas med stora bokstäver),