Dagens föreläsning

- Informellt om klassen NP
- Formellt om klassen NP
- NP versus P
- (Om tid finns) repetition av tidigare ämnen.

Några svåra problem

- Att vi kan lösa en del problem i polynomial tid beror ofta på att vi kan hitta en smartare metod än ren rå-styrka (dvs. något bättre än att prova alla möjliga lösningar).
- Tyvärr är inte det sant för alla avgörbara problem ...
- Det kryllar faktiskt av problem som vi helt enkelt inte vet om de kan avgöras i polynomial tid. Exempel är
  - The quadratic assingment problem
  - Satisfierbarhet (SAT)
  - Kappsäcksproblemet
  - Att hitta en s.k. Hamiltonväg i en riktad graf

The quadratic assignment problem

Givet: n tomter och lika många fabriker, samt en förteckning över hur mycket resurser som måste transporteras dagligen mellan varje par av fabriker.

Frågan är: kan man bygga fabrikerna på dessa tomter (precis en fabrik per tomt) så att de dagliga transporterna k ton-mil.

Som ett språk:

$$\mathsf{QAP}_k = \{\langle \text{tomter}, \text{fabriker}, \text{transportbehov} \rangle \mid \text{Det finns ett sätt att byga fabrikerna så att...}\}$$

Satisfierbarhet (SAT)

Givet: En boolsk formel, till exempel

$$\varphi = (\neg A \vee B) \wedge (\neg B \vee C) \wedge (\neg C \vee \neg B) \wedge A \ .$$

Frågan är: Kan de ingående variablerna (i det här fallet A, B, och C tilldelas sanningsvärden (dvs. sant eller falskt) så att formeln som helhet blir sann? Som ett språk:

$$\mathsf{SAT} = \{\langle\varphi\rangle \mid \text{Det finns ett sätt att tilldela sanningsvärden till de ingående variablerna ...}\}$$

Kappsäcksproblemet

Givet: En stor kappsäck och ett antal föremål att packa med olika värden och volym.

Frågan är: Kan man packa kappsäcken så att det totala innehållet blir värt k kronor?

Som ett språk:

$$\mathsf{KNAPSACK}_k = \{\langle \text{kappsäckens storlek}, \text{en lista med föremål} \rangle \mid \ldots\}$$

Hamiltonväg

Givet: En graf G och två noder s och t i G.

Frågan är: Finns det en väg från s till t som passerar alla noder i G exakt en gång?

Som ett språk:

$$\mathsf{HAMPATH} = \{\langle G, s, t\rangle \mid \text{Det finns en väg från } s \text{ till } t \text{ som } \ldots\}$$

Verifiering

- Något som dessa problem har gemensamt är att vi kan verifiera om en given hypotes är en korrekt lösning i polynomial tid.
- Med andra ord, om man vet att ett element tillhör något av de här språken, är det lätt att övertyga andra, men så är det inte alltid.

Definition av verifierare

Definition 7.18 En *verifierare* för ett språk A är en algoritm V, där

$$A = \{w \mid V \text{ accepterar } \langle w, c\rangle \text{ för någon sträng } c\}$$

Vi mäter tiden det tar för en verifierare att arbeta i termer av längden av w, så en *polynomialstids verifierare* exekverar i polynomial tid med avseende på längden av w. Ett språk är *polynomialtids verifierbart* om det är har en polynomialtids verifierare. Strängen c är ett *certifikat* för w.

Definition av NP

Definition 7.19 NP är klassen av språk som har polynomialtidsverifierare.

En ekvivalent definition är klassen av språk som avgörs av en ickedeterministisk polynomialtids TM. Kan ni se varför?

En NTM som avgör **HAMPATH**

$V =$ "På input $\langle G, s, t \rangle$, där G är en riktad graf med noderna s och t.

1. Skriv en lista med m nummer, $p_1, \ldots, p_m$, där m är antalet noder i G. Varje nummer väljs ickedeterministiskt till ett tal mellan 1 och m.
2. Kolla om $p_1 = s$ och $p_m = t$, om inte, *refusera*.
3. För varje i mellan 1 och $m - 1$, kolla om (p_i, p_{i+1}) är en kant i G. Om inte, *refusera*. När hela listan gåtts igenom, *acceptera*."

En till definition av NP

Teorem 7.20 Ett språk är i NP om och endast om det avgörs av en ickedeterministisk TM.

Vi visar hur man konverterar en polynomialtids verifierare till en ekvivalent polynomialtids NTM och vice versa. NTM:en simulerar verifieraren genom att gissa certifikatet, och verifieraren simulerar NTM:en genom att använda den accepterande grenen av NTM:ens beräkning som certifikat. Vi tittar närmare på resonemanget.

Låt A vara ett språk i NP, och låt V vara dess polynomialtidsverifierare. Vi antar att V är en TM som exekverar på tid n^k och konstruerar en NTM N som följer.

$N =$" På input w av längd n:

1. Gissa en sträng c som inte är längre än n^k.
2. Kör V på input $\langle w, c \rangle$.
3. Svara som V. (obs! V behöver inte stanna)"

Antag att A är ett språk som avgörs av en NTM N i polynomial tid. Vi kan då konstruera en polynomialtidsverifierare V som följer.

$V =$“ På input $\langle w, c \rangle$ där w och c är strängar:

1. Simulera N på input w, och låt c bestämma hur N ska fortsätta när det finns fler än ett alternativ.
2. Svara som den här grenen av N’s beräkning.”

Ickedeterministisk tid

Definition 7.21

$$\mathsf{NTID}(t(n)) = \{L \mid L \text{ avgörs av en } O(t(n)) \text{ tids ickedeterministisk TM}\}$$

Detta innebär att

$$\mathsf{NP} = \cup_k \mathsf{NTID}(n^k) \ .$$

Fråga 2

En *clique* i en oriktad graf är komplett delgraf. En clique av storlek k är helt enkelt en clique som innehåller k noder.

Är problemet att avgöra om en graf har en clique av storlek k i NP?

P versus NP

- P är klassen av alla språk som kan avgöras i polynomial tid med en deterministisk TM.
- NP är klassen av alla språk som kan avgöras i polynomial tid med en ickedeterministisk TM.

Alternativt,

- NP är klassen av alla språk som kan verifieras i polynomial tid med en deterministisk TM.

Nu är frågan, är $\mathsf{P} = \mathsf{NP}$?