

Linköpings Universitet
Institutionen för datavetenskap
Peter Jonsson

TDDA32 Konstruktion och analys av algoritmer

Övningsuppgifter 3

NP-fullständighet.

Uppgift 1.

Visa att följande problem är NP-fullständigt.

Instans: Mängd S , en mängd $\mathcal{C} = \{C_1, \dots, C_m\}$ där varje $C_i \subseteq S$ samt ett heltal $K \geq 0$.

Fråga: Finns en mängd $\mathcal{C}' \subseteq \mathcal{C}$ sådan att $|\mathcal{C}'| \leq K$ och $\bigcup \mathcal{C}' = S$?

Tips: Reducera från nodhöljesproblemet.

Uppgift 2*.

Visa att följande problem (som brukar kallas *Subset Sum*) är NP-fullständigt:

Instans: En ändlig mängd positiva heltal S och ett positivt heltal t .

Fråga: Finns en mängd $S' \subseteq S$ sådan att $\sum S' = t$?

Tips 1: Reduktion från 3SAT. Tips 2: Bevis finns i kursboken (Th. 34.15).

Uppgift 3.

Visa att följande problem (som brukar kallas *Knapsack*) är NP-fullständigt:

Instans: En mängd objekt $O_1, \dots, O_n$ vilka har vikter a_i , $1 \leq i \leq n$ och värden v_i , $1 \leq i \leq n$. Två heltal B och K .

Fråga: Finns en delmängd av $O_1, \dots, O_n$ sådan att deras sammanlagda vikt är $\leq B$ och deras sammanlagda värde $\geq K$.

Tips: Reduktion från Subset Sum.

Inexakta metoder.

Uppgift 4.

Låt V vara en mängd variabler och C en mängd villkor av typen $(x < y)$ där x, y är distinkta variabler i V . Vi säger att (V, C) har en lösning om och endast om det existerar en funktion $f : V \rightarrow \mathbb{R}$ sådan att för alla $(x < y) \in C$ gäller $f(x) < f(y)$. Konstruera en polynomisk algoritm som beräknar en mängd $C' \subseteq C$ med följande egenskaper:

1. (V, C') har en lösning; och
2. $|C'| \geq |C|/2$.

Algoritmen får gärna vara randomiserad och då ändras krav 2 till följande:

- 2' den *förväntade* storleken på C' skall vara $\geq |C|/2$.

Uppgift 5.

Maximeringsproblemet MAX DICUT definieras så här:

Indata: Riktad graf $G = (V, E)$ som inte innehåller någon självloop (d.v.s. saknar bågar av typen (v, v)).

Lösning: Två mängder V_1, V_2 sådana att $V = V_1 \cup V_2$ och $V_1 \cap V_2 = \emptyset$.

Mått: Antal bågar som har sin startpunkt i V_1 och sin slutpunkt i V_2 .

Konstruera en polynomisk algoritm som approximerar MAX CUT inom 4, d.v.s. hittar en lösning som har ett mått på minst 25% av den optimala lösningen. Algoritmen får vara randomiserad och då är kravet att den *förväntade* storleken på lösningen är minst 25% av optimum.