Cvičenie 6

Príklad 6.1.

Máte mince s hodnotami $c_1, \ldots, c_k$. Nájdite spôsob vydania sumy n pomocou najmenšieho počtu mincí.

Ako sa riešenie zmení, ak by boli hodnoty c_i navzájom rôzne a každého z týchto "typov" mincí by sme mali neobmedzene veľa?

Náčrt riešenia 6.1.

Budeme sa snažiť vyriešiť nasledovný problém, ktorý si označíme $F(i, j)$: Koľko najmenej z prvých i mincí musím použiť, aby som zaplatil sumu j?

Mali by sme vidieť jednoduchú rekurzívnu závislosť

$$F(i, j) = \min(F(i-1, j), F(i-1, j-c_i) + 1)$$

buď i-tu mincu nepoužijeme (preto sa nám nezmení j) alebo ju použijeme, preto k výsledku pripočítavame 1. Časová zložitosť takéhoto riešenia je $O(nk)$.

Ak máme z každej mince ľubovoľne veľa kusov, jediné čo musíme spraviť je dovoliť programu, aby tú istú mincu vybral viackrát. Keďže na poradí, v akom vyberáme mince nám nezáleží, môžeme sa rozhodovať nasledovne: buď mincu vyberieme a môžeme ju vybrať aj v ďalšom kroku, alebo ju nevyberieme, ale potom ju už nikdy viac vybrať nemôžeme.

$$F'(i, j) = \min(F'(i-1, j), F'(i, j-c_i) + 1)$$

Príklad 6.2.

Na vstupe máte pole celých kladných čísel. Vyberte si niekoľko z nich tak, aby žiadne dve čísla neboli susedné, ich súčet bol deliteľný 13 a zároveň bol tento súčet čo najväčší možný.

Náčrt riešenia 6.2.

Dynamické programovanie je v tomto prípade pomerne priamočiare. Najjednoduchšie sa nám nad oboma podmienkami bude uvažovať zvlášť a tieto riešenia potom spojíme. V oboch prípadoch budeme ako menší podproblém uvažovať kratšiu postupnosť, teda sa budeme pýtať akú najväčšiu hodnotu vieme docieliť z prvých i prvkov postupnosti.

Zaručiť to, že susedné prvky nie sú vybrané vieme jednoducho pomocou spôsobu, akým rekurziu skonštruujeme – ak posledný prvok vyberieme, musíme sa zavolať na podproblém o jedno kratší.

$$F(i) = \max(F(i-1), F(i-2) + c_i)$$

Deliteľnosť 13 vieme naopak schovať do stavov nášho dynamického programovania, ktoré si rozšírime o zvyšok po delení. Budeme sa teda pýtať, aký najväčší súčet vieme dosiahnuť z prvých i prvkov tak, aby jeho zvyšok po delení bol x.

$$F'(i, x) = \max(F'(i-1, x), F'(i-1, (x - c_i) \mod 13) + c_i)$$

Nakoniec spojiť tieto dve funkcie tak, aby platili obe podmienky už určite zvládnete.

Príklad 6.3.

Problém kliky je grafový problém, v ktorom chceme zistiť, či v zadanom grafe existuje úplný podgraf s k vrcholmi.

Dokážte, že problém kliky je NP-úplný.

Náčrt riešenia 6.3.

Dôkaz nájdete v skriptách na stranách 42 a 43.

Príklad 6.4.

Dokážte, že platí $3 - SAT \leq_P SUBSET - SUM$

$\leq_P$ označuje polynomiálnu redukciu.

Problém $SUBSET - SUM$ znie nasledovne: Existuje podmnožina čísel $s_1 \dots s_n$ so súčtom k?

Náčrt riešenia 6.4.

Pre každú premennú vytvoríme dve čísla (pre kladnú hodnotu a pre zápornú). Donútime vybrať vždy práve jedno z týchto čísel.

Nech je daná CNF $(a_{1,1} \vee a_{1,2} \vee a_{1,3}) \wedge \dots \wedge (a_{n,1} \vee a_{n,2} \vee a_{n,3})$, ktorá obsahuje m premenných. Pre každú premennú u_i vytvoríme čísla v_i a v_i' s $m+n$ číslicami v desiatkovej sústave:

	u_1	u_2	$\dots$	u_i	$\dots$	u_m	$C_i \dots C_n$
v_i (zodpovedá u_i)	0	0	$\dots$	1	$\dots$	0	$C_k = 1$ ak u_i je v k-tej klauzule
v_i' (zodpovedá $\neg u_i$)	0	0	$\dots$	1	$\dots$	0	$C_k = 1$ ak $\neg u_i$ je v k-tej klauzule

Platí, že ak vyberieme nejakú podmnožinu, ktorá zodpovedá spĺňajúcemu priradeniu, tak ich súčet bude mať podobu

u_1	u_2	$\dots$	u_m	C_1	C_2	$\dots$	C_n
1	1	$\dots$	1	≥ 1	≥ 1	$\dots$	≥ 1
				≤ 3	≤ 3	$\dots$	≤ 3

Ako cieľ si zvolíme číslo t:

u_1	u_2	$\dots$	u_m	C_1	C_2	$\dots$	C_n
1	1	$\dots$	1	4	4	$\dots$	4

Pre každú klauzulu pridáme doplnkové čísla pre prípad, že stĺpec danej klauzuly bude mať menej než 4 v súčte:

	u_1	u_2	$\dots$	u_m	C_1	C_2	$\dots$	C_i	$\dots$	C_n
s_i	0	0	$\dots$	0	0	0	$\dots$	1	$\dots$	0
s_i'	0	0	$\dots$	0	0	0	$\dots$	2	$\dots$	0