ALG 11

Dynamické programování

Úloha batohu neomezená

Úloha batohu 0/1

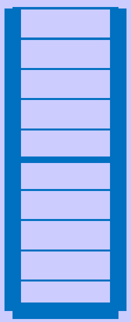
Úloha batohu / Knapsack problem

Máme N předmětů, každý s váhou Vi a cenou Ci (i = 1, 2, ..., N)
a batoh s kapacitou váhy K.
Máme naložit batoh těmito předměty tak,
aby kapacita K nebyla překročena a obsah měl maximální cenu.

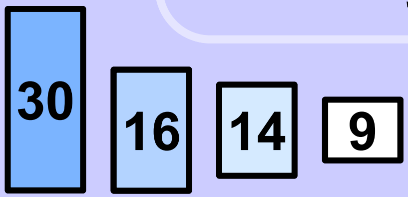
Neomezená varianta -- Každý předmět lze použít libovolněkrát.

0/1 varianta -- Každý předmět lze použít nejvýše jednou.

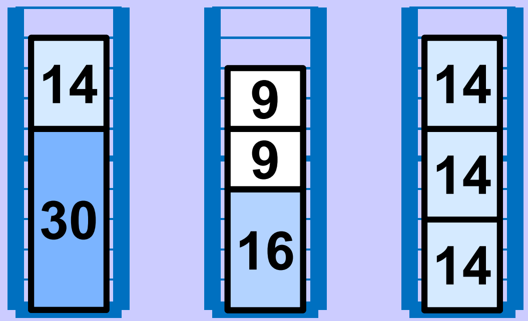
Schematický batoh
s kapacitou 10



Předměty
s uvedenou
cenou,
váha ~ výška



Několik možných
konfigurací

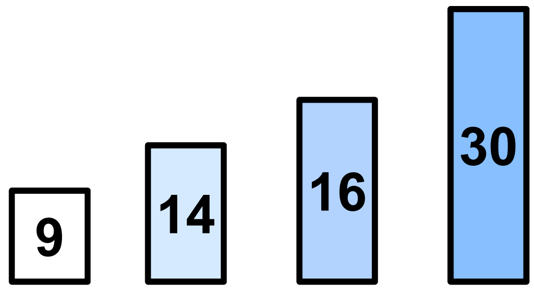


Neomezená úloha batohu

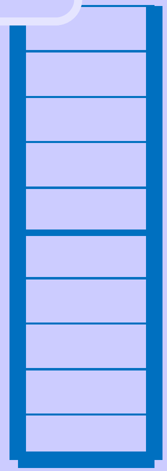
Příklad

N = 4

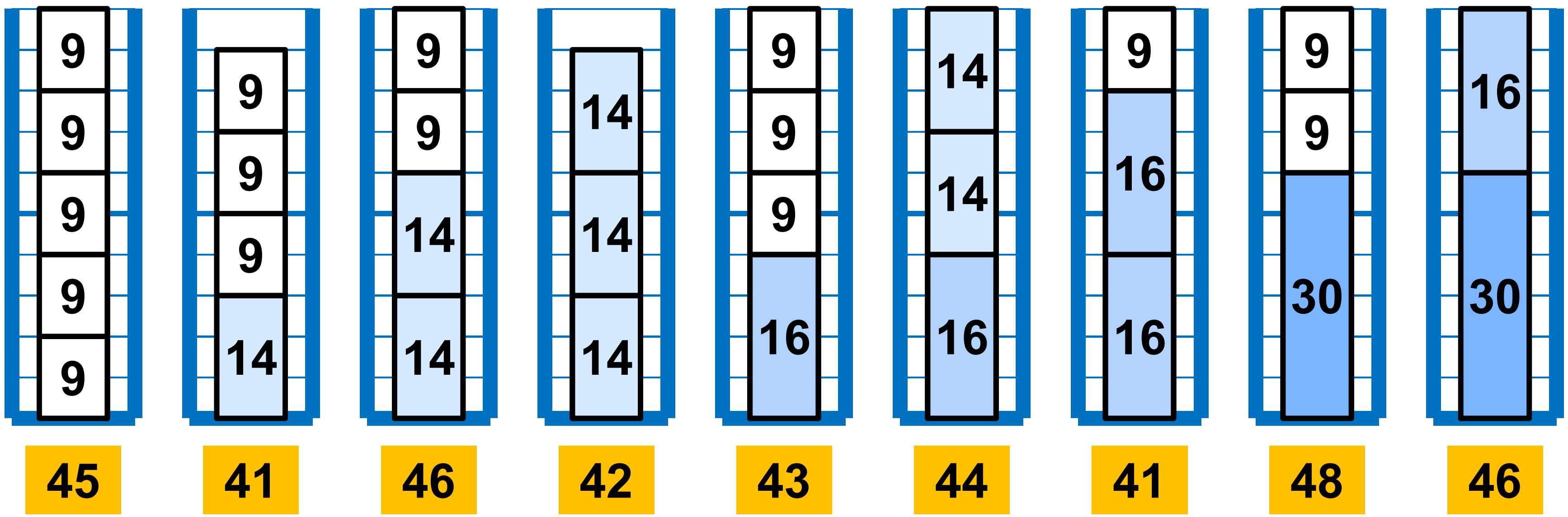
Váha	2	3	4	6
Cena	9	14	16	30



Batoh s kapacitou 10



Některé možnosti naplnění a odpovídajíci ceny



Neomezená úloha batohu

Použijeme K+1 batohů, o kapacitách 0, 1, 2, 3, ..., K.
Hodnotu optimálního naplnění batohu s kapacitou K
lze získat jako maximum z hodnot

-- (optimalní naplnění batohu o kapacitě K - V1) + C1,
-- (optimalní naplnění batohu o kapacitě K - V2) + C2,
...
-- (optimalní naplnění batohu o kapacitě K - VN) + CN.

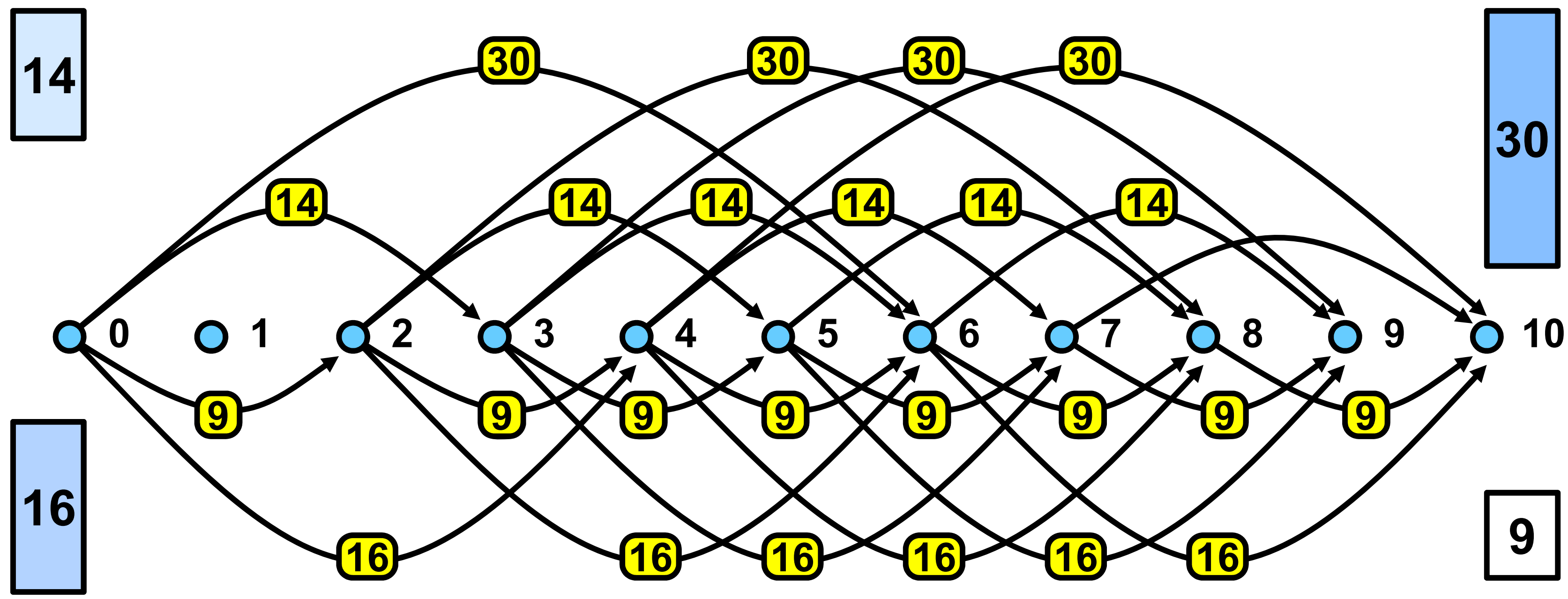
Optimální naplnění batohu o kapacitě K - Vi (i = 1..N) je stejnou úlohou, jen s menšími daty. Hodnoty předpočítáme standardně metodou DP do 1D tabulky.

Neomezenou úlohu batohu lze přímo vyjádřit jako úlohu nalezení nejdelší cesty v DAG. Postup řešení je identický.

Neomezená úloha batohu -- převod na DAG

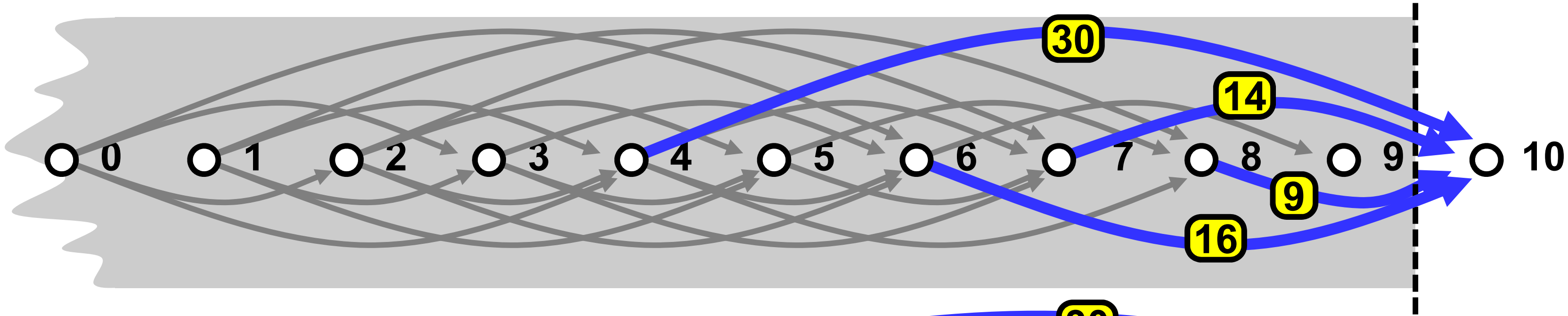
DAG:
Uzly: Kapacity 0, 1, 2, 3, ..., N.
Hrany: Z uzlu X vedou hrany po řadě do uzlů X+V1, X+V2, ..., V+VN, jsou po řadě ohodnoceny cenami C1, C2, ..., CN.

Příklad K = 10, N = 4, Vi = (2, 3, 4, 6), Ci = (9, 14, 16, 30), i = 1..4.



Neomezená úloha batohu -- jako DAG

Optimální naplnění kapacity 10 = ??



Opt. naplnění kapacity 4 = 18 — 4 5 6 7 8 9 10 (30)

Optimální naplnění kapacity 6 = 30 — 6 7 8 9 10 (16)

Optimální naplnění kapacity 7 = 32 — 7 8 9 10 (14)

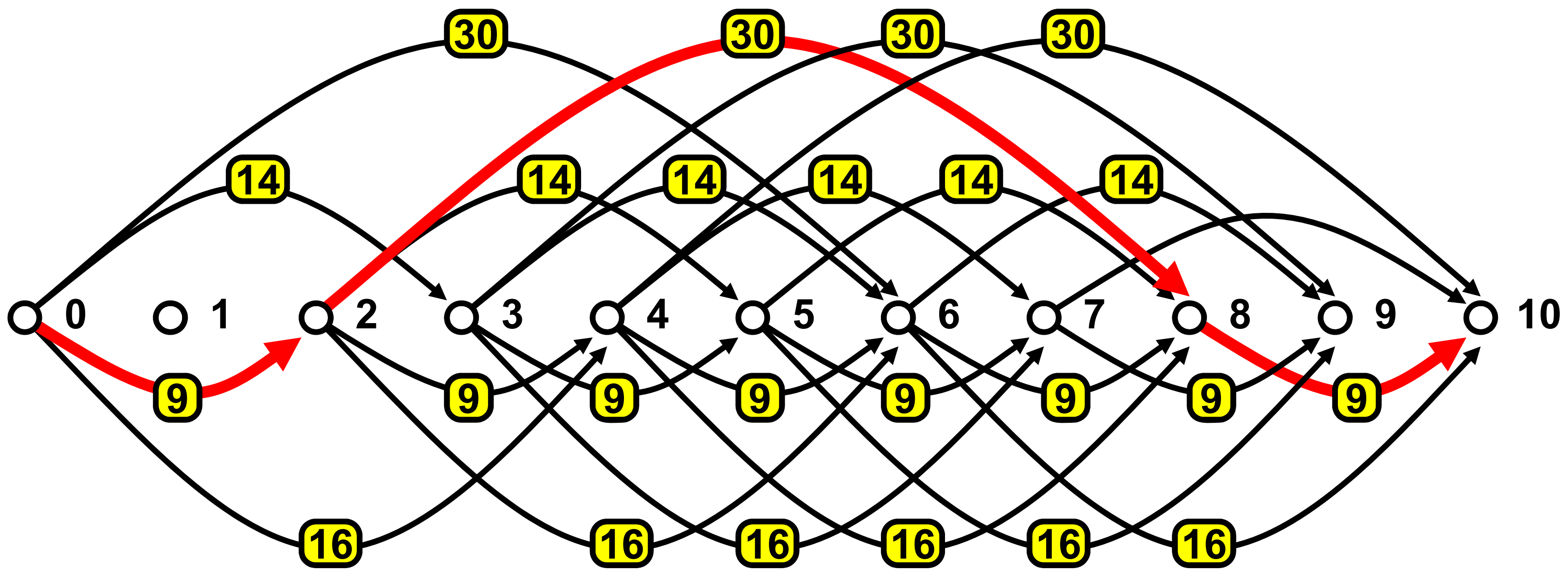
Optimální naplnění kapacity 8 = 39 — 8 9 10 (9)

Optimální naplnění kapacity 10 = max(18 + 30, 30 + 16, 32 + 14, 39 + 9) = 48

Neomezená úloha batohu

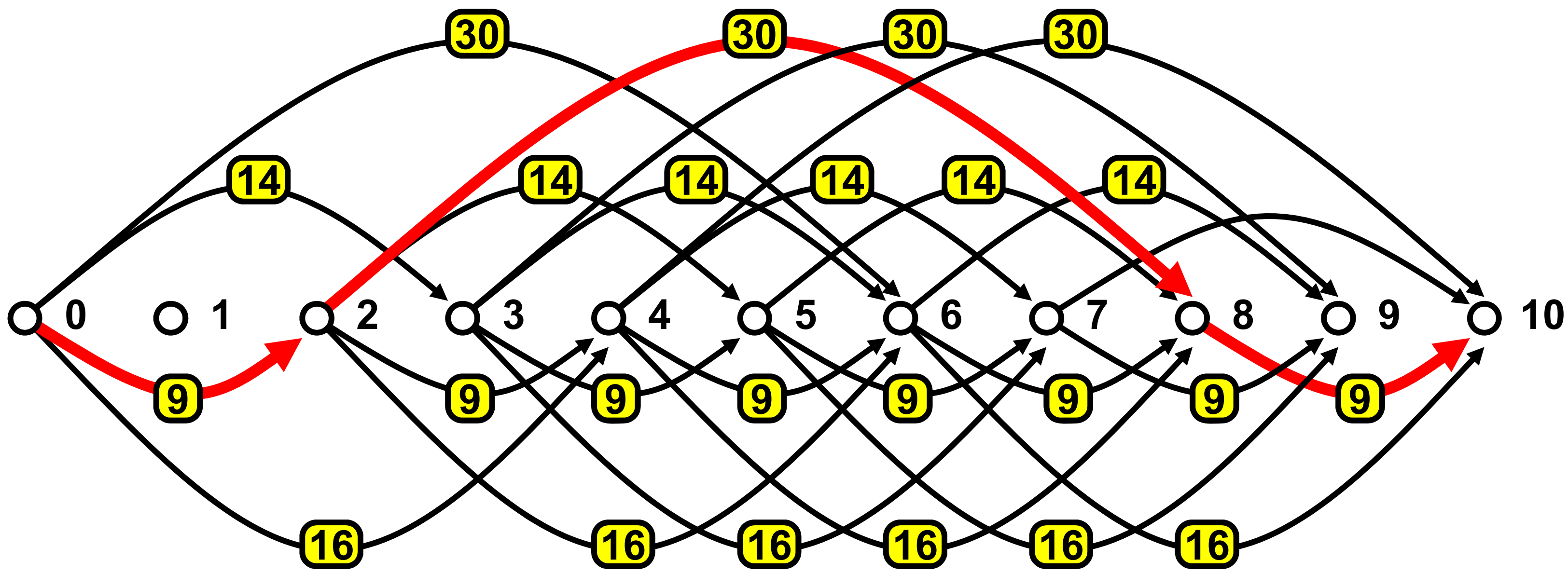
Nejdelší cesta odpovídá optimálnímu naplnění batohu.
Dvě hrany s cenou 9 a jedna hrana s cenou 30, celkem cena = 48.

Batoh optimálně naplníme dvěma předměty s váhou 2 a cenou 9 a jedním předmětem s váhou 6 a cenou 30.



Neomezená úloha batohu -- asymptotická složitost

DAG obsahuje K+1 uzlů a méně než K*N hran.
Má tedy V = Θ(K) uzlů a E = O(K*N) hran.
Asymptotická složitost hledání nejdelší cesty je Θ(V+E),
máme tedy pro neomezenou úlohu batohu
asymptotickou složitost O(K + K*N) = O(K*N).



Neomezená úloha batohu -- Asymptotická složitost

Zdánlivá nesrovnalost

1. Literatura: NP těžký problém, není znám efektivní algoritmus.
2. ALG OI: DP řeší úlohu v čase v O(N*K), tedy efektivně?

Délka výpočtu DP je lineárně závislá na velikosti kapacity K.

Příklad
Velkou kapacitu 2^{64} lze zadat velmi krátkým zápisem
Kapacita = 18446744073709551616.
N = 3. Položky (váha, cena): (2, 345), (3, 456), (5, 678).
Data lze zapsat do cca 100 bitů < 16 Bytů < "dva longy"
Výpočet pomocí DP potrvá přes 584 roky
za předpokladu, že za 1 sec vyplní 10^9 prvků tabulky.

Délka výpočtu DP je exponenciálně závislá na délce řetězce definujícího kapacitu K.

0/1 úloha batohu

Každý předmět lze použít nejvýše 1 krát.

Máme vybrat vhodnou podmnožinu předmětů splňující zadání úlohy. Každé podmnožině lze přiřadit charakteristický vektor z hodnot 0/1 délky N. Pozice ve vektoru odpovídá předmětu, 0 resp. 1 odpovídá nepřítomnosti resp. přítomnosti předmětu v této podmnožině. Binárních vektorů délky N je celkem 2^N, systematické probírání všech možných podmnožin bude mít exponenciální asymptotickou složitost, nehodí se.

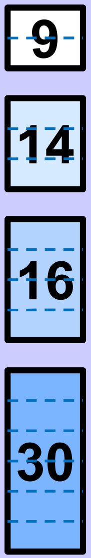
DP poskytuje (pro relativně nevelké kapacity) výhodnější postup.

0/1 úloha batohu

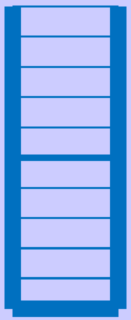
Příklad

N = 4

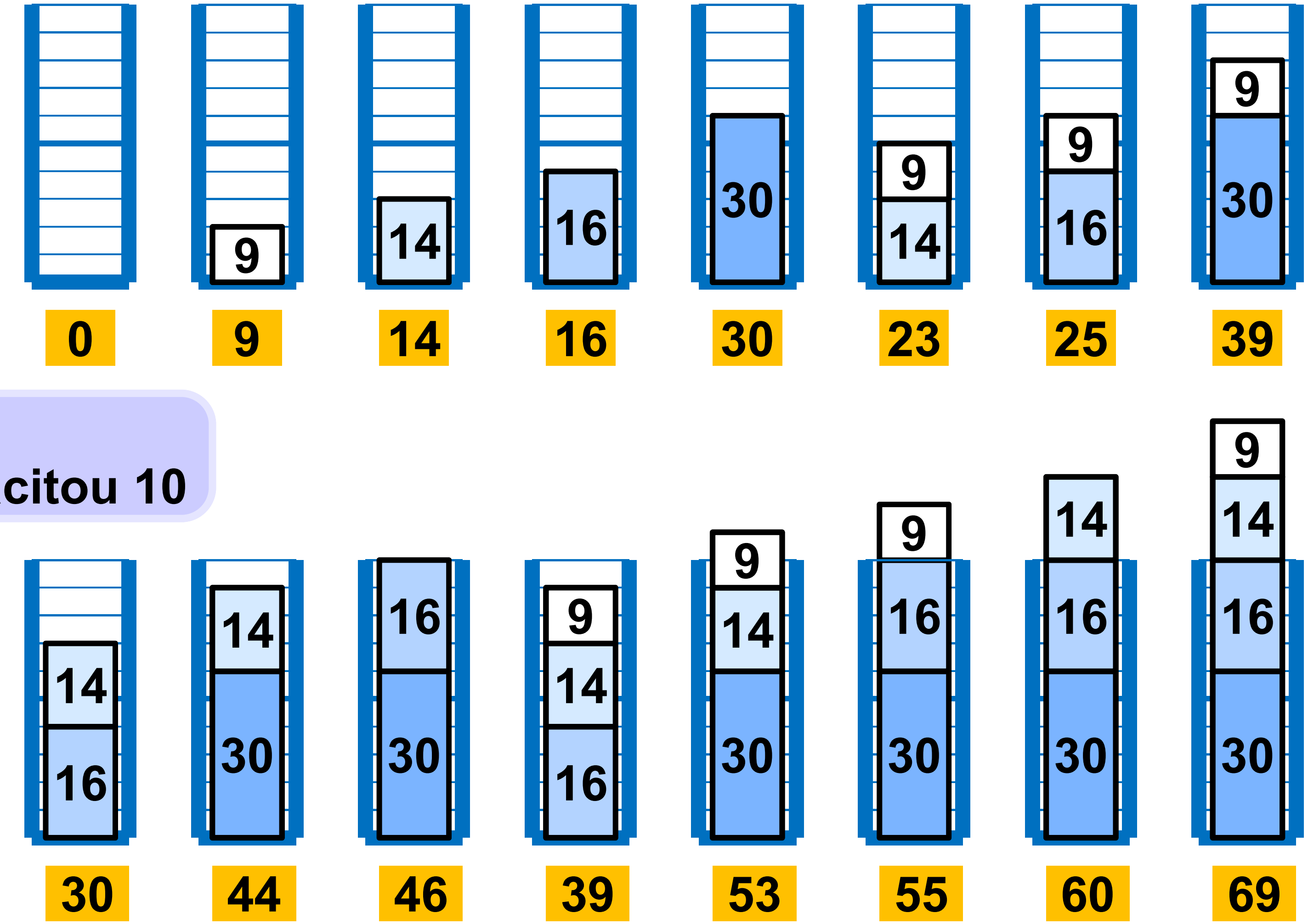
Váha	Cena
2	9
3	14
4	16
6	30



Batoh
s kapacitou 10



Všech 16 podmnožin čtyř předmětů a jejich ceny



0/1 úloha batohu -- řešení

Použijeme K+1 batohů, o kapacitách 0, 1, 2, 3, ..., K.
Použijeme N+1 souborů předmětů.

Soubor 0 neobsahuje žádný předmět.
Soubor 1 obsahuje předmět 1.
Soubor 2 obsahuje předměty 1 a 2.
Soubor 3 obsahuje předměty 1, 2, 3.
...
Soubor N obsahuje předměty 1, 2, 3, ..., N.

Na pořadí předmětů nezáleží, je ale zafixované.

Pro každou kapacitu a pro každý soubor budeme řešit
stejnou úlohu metodou DP, v pořadí od menších hodnot k větším.

0/1 úloha batohu -- řešení

Označme symbolem U(x, y) úlohu
se souborem předmětů 1, 2, ..., x a s kapacitou batohu y
a symbolem Opt(x, y) optimální řešení této úlohy.

Pro řešení U(x,y) použijeme optimální řešení úloh U(x-1, _).

Buď do Opt(x, y) zahrneme předmět x nebo jej nezahrneme.
V prvním případě použijeme hodnotu řešení pro batoh
s kapacitou menší o velikost váhy Vx, tedy hodnotu Opt(x-1, y-Vx).
V druhém případě beze změny použijeme hodnotu Opt(x-1, y).

Z obou hodnot vybereme tu výhodnější a dostáváme tak:

Opt(x, y) = max(Opt(x-1, y), Opt(x-1, y-Vx)).
Dále zřejmě platí Opt(0, y) = Opt(x, 0) = 0, pro x = 0..N, y = 0..K.

0/1 úloha batohu -- řešení

Pro x = 1..N, y = 0..K:

Opt(x, y) = max(Opt(x-1, y), Opt(x-1, y-Vx)).
Opt(0, y) = Opt(x, 0) = Opt(0, 0) = 0.

Pokud y-Vx < 0, položíme Opt(x, y-Vx) = $-\infty$ (a netabelujeme).

Hodnoty Opt(x,y) tabelujeme ve 2D tabulce velikosti (N+1)$\times$(K+1)
s řádkovým indexem x (předměty)
a sloupcovým indexem y (kapacity menších batohů).

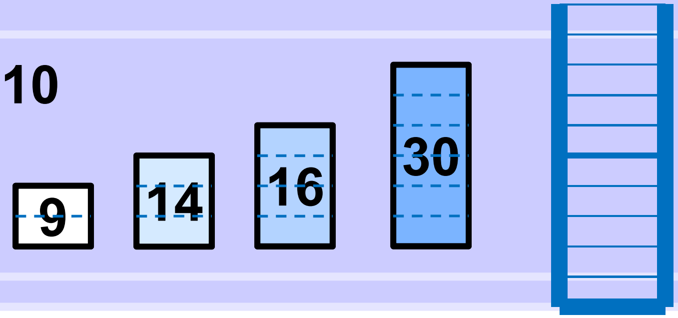
Pro rekonstrukci optimálního řešení použijeme tabulku
předchůdců stejné velikosti jako tabulku pro Opt(x, y).
Předchůdce leží vždy v předchozím řádku x-1, stačí registrovat
buď pozici y (beze změny) nebo y-Vx (přidán předmět x).

0/1 úloha batohu

Příklad

N = 4 Kapacita = 10

Váha	2	3	4	6
Cena	9	14	16	30



Opt(x, y)

	0	1	2	3	4	5	6	7	8	9	10
0	0	0	0	0	0	0	0	0	0	0	0
1	0	0	9	9	9	9	9	9	9	9	9
2	0	0	9	14	14	23	23	23	23	23	23
3	0	0	9	14	16	23	25	30	30	39	39
4	0	0	9	14	16	23	30	30	39	44	46

Pred(x, y)

	0	1	2	3	4	5	6	7	8	9	10
0	--	--	--	--	--	--	--	--	--	--	--
1	0	1	0	1	2	3	4	5	6	7	8
2	0	1	2	0	1	2	3	4	5	6	7
3	0	1	2	3	0	5	2	3	4	5	6
4	0	1	2	3	4	5	0	7	2	3	4

0/1 úloha batohu

Vyjádření jako optimální cesty v DAG

Uzly DAG budou jednotlivé hodnoty Opt(x, y), x = 0..N, y = 0..K, celkem bude mít DAG (N+1)*(K+1) uzlů.
Do uzlu Opt(x, y) povede hrana

-- Opt(x-1, y) --> Opt(x, y)
ohodnocená 0 (žádný přidaný předmět),

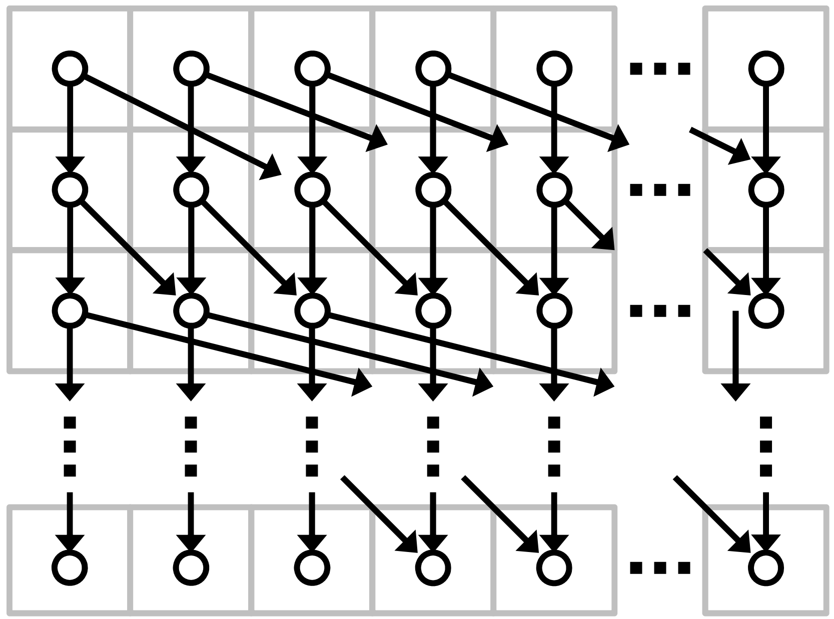
-- a pokud y-Vx ≥ 0, také hrana
Opt(x-1, y-Vx) --> Opt(x, y)
ohodnocená cenou Cx (cenou přidaného předmětu x).

V takto zkonstruovaném DAG hledáme nejdelší (= nejcennější) cestu standardní DP metodou.

Jaké je topologické uspořádání tohoto DAG?

0/1 úloha batohu - topologické uspořádání DAG

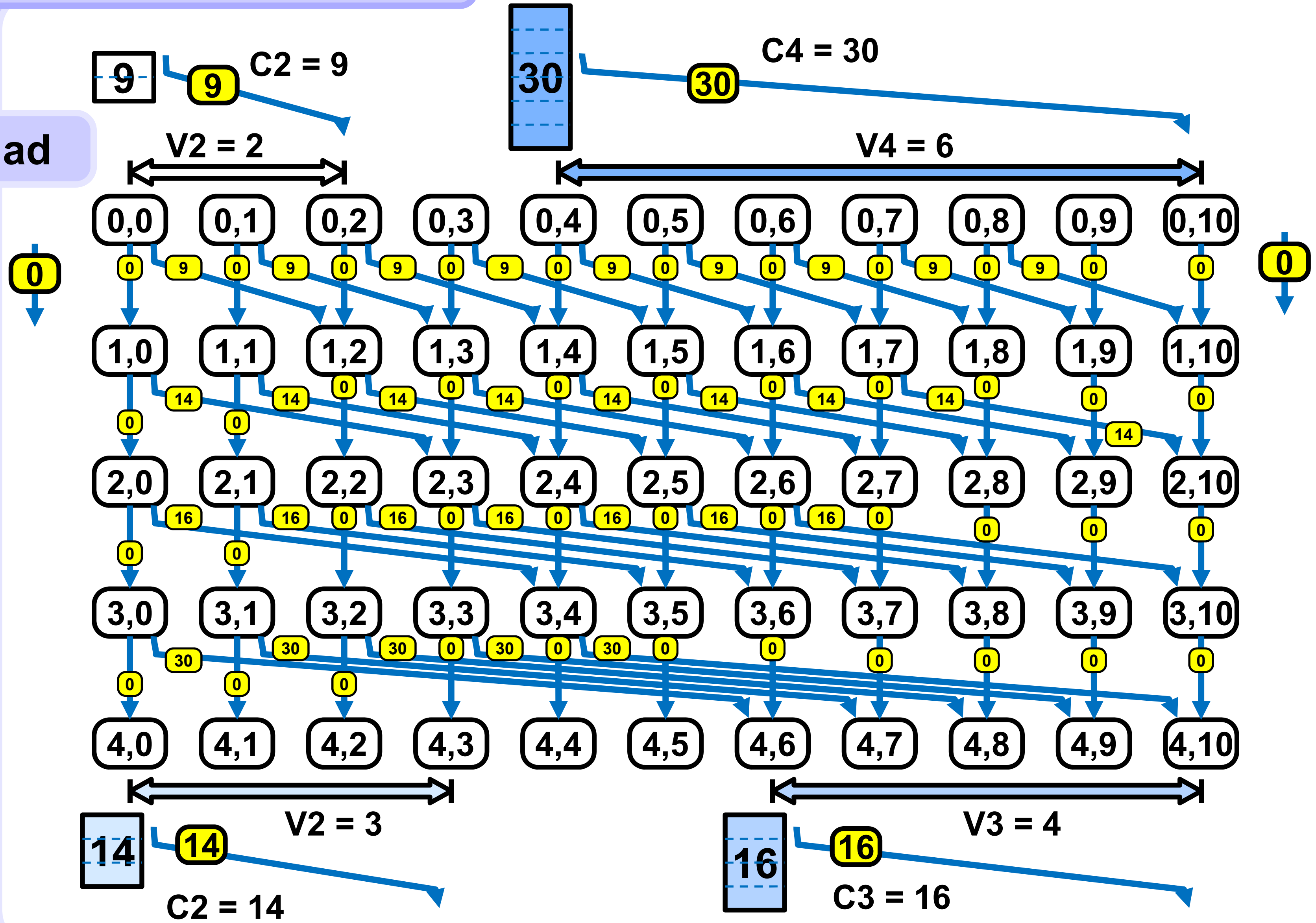
DAG můžeme uvažovat nakreslený formálně do DP tabulky, přičemž uzel Opt(x, y) leží v buňce s indexy x a y. Pak hrany DAG vedou vždy pouze z předchozího řádku do následujícího řádku. Pokud tento DAG procházíme shora po řádcích, to jest ve stejném pořadí, v němž vyplňujeme DP tabulku, respektujeme jeho topologické uspořádání.



V tomto případě není nutno uzly DAG v topologickém uspořádání uvažovat v jedné přímce, "tabulkové" uspořádání je přehlednější.

0/1 úloha batohu -- DAG

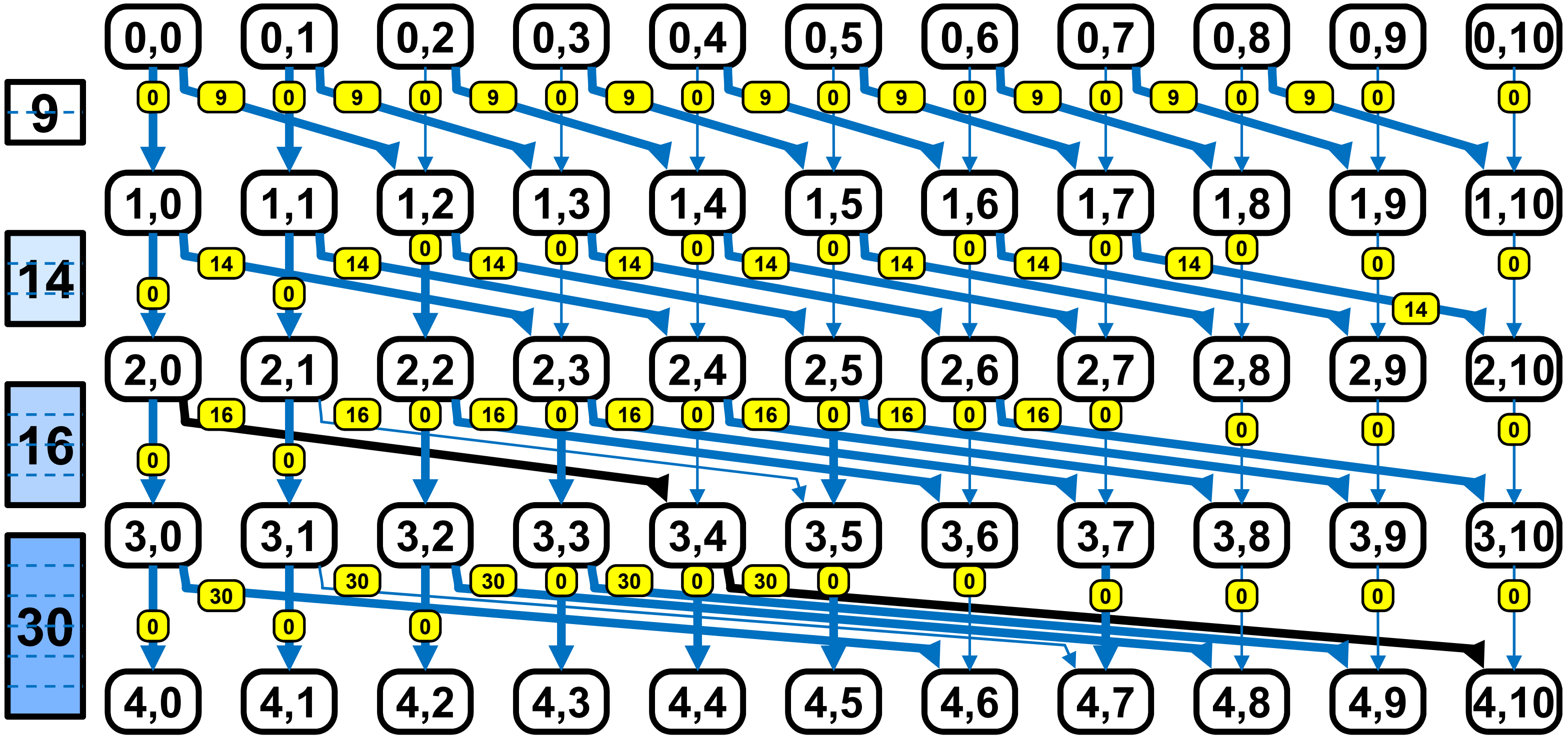
Příklad



0/1 úloha batohu -- rekonstrukce optimálního řešení pomocí tabulky předchůdců

Pred(x, y)

	0	1	2	3	4	5	6	7	8	9	10
0	--	--	--	--	--	--	--	--	--	--	--
1	0	1	0	1	2	3	4	5	6	7	8
2	0	1	2	0	1	2	3	4	5	6	7
3	0	1	2	3	0	5	2	3	4	5	6
4	0	1	2	3	4	5	0	7	2	3	4



0/1 úloha batohu

Asymptotická složitost

Tabulka ... Velikost ... $(N+1)*(K+1) \in \Theta(N*K)$
Vyplnění jedné buňky ... $\Theta(1)$
Vyplnění tabulky ... $\Theta(N*K*1) = \Theta(N*K)$.
Rekonstrukce optimálního řešení $\Theta(N)$.
Celkem ... $\Theta(N*K + N) = \Theta(N*K)$.

DAG .... Uzlů ... $(N+1)*(K+1) \in \Theta(N*K)$.
Hran ... nejvýše $2*(N+1)*(K+1)$, tj $\in O(N*K)$.
Nalezení optimální cesty ... Θ(uzlů+hran) = $\Theta(N*K)$.

Řešení obou variant úlohy batohu, neomezené i 0/1,
má asymptotickou složitost $\Theta(N*K)$.

Přitom zároveň platí:

Asymptotická složitost DP řešení je exponenciální vzhledem k délce řetězce definujícího kapacitu K.