Knapsack

volume

$V \in \mathbb{Z}$

Имаме раница с обем V и предмети като всеки предмет i има размер s_i и стойност c_i ($s_i \in \mathbb{Z}^+$, а $v_i \in \mathbb{R}^+$).

size

value

Целта ни е да изберем такова подмножество на предметите, т.е. да не надвишим обема на раницата и в същото време да максимизираме ст-та на всички предмети в нея.

Вариант 1: Имаме неограничен брой от всеки предмет. (позв. неогран. повторения)

Вариант 2: Имаме краен брой от всеки предмет. (може този краен брой като отделни различни предмети да си ги запишем, т.е без неогранич. повторения).

V

$A[1,\dots,n]$

$S[1,\dots,n]$

$C[1,\dots,n]$

① #подпроблемите

$A[1,\dots,i]$

$C[1,\dots,i]$

$S[1,\dots,i]$

раница със обем $0 \le V' \le V$, $V' \in \mathbb{Z}^+$

Дали от масива $A[1,\dots,i]$ можем да изберем подредица $A[i_1],\dots,A[i_k]$, $\{i_1,\dots,i_k\} \subseteq \{1,\dots,i\}$

т.е. $\sum_{j\in\{i_1,\dots,i_k\}} S[j] \le V'$

$\sum_{j\in\{i_1,\dots,i_k\}} C[j] \to \max$

$\Theta(n.V)$

② Избор (решаване на подпроблем).

За всеки предмет $A[j]$ $1\le j \le i$ се питаме:

$\Theta(1)$

- ако $S[j] \le V'$, можем да го добавим, ако искаме
- ако $S[j] > V'$ или не го искаме този предмет, просто няма да го добавим в раницата.

③ $dp[0,\dots,n][0,\dots,V]$ като

$dp[i][j]$ = макс. ст-т на решенията с обем j
и избор на подмножество от
предмети от $A[1,\dots,i]$

$dp[0][0] \leftarrow 0$

$dp[i][0] \leftarrow 0, \ 1 \le i \le n$

$dp[0][j] \leftarrow 0, \ 1 \le j \le V$

for $i \leftarrow 1$ to n do:
 for $j \leftarrow 1$ to V do:
  $dp[i][j] \leftarrow \max\{\underbrace{dp[i-1][j-S[i]] + C[i]}_{\text{included}},\ \underbrace{dp[i-1][j]}_{\text{exclude}}\}$

included → if $S[i] \le j$

$dp[n][V]$
е решението ни.

$i \qquad A[1,\dots,i]$

$1 \le i_1 < i_2 < \dots < i_k \quad \{i_1,\dots,i_k\} \subseteq \{1,\dots,i\}$

$\boxed{HW}$ Помислете си вариант 1 на задачата.

Приложение

① От ДП

Имате $n > 0$ купчинки като във всяка от тях имате $a_i > 0$ жетони.

Търсим $I = \{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$ т.е.

$$\sum_{i \in I} a_i \to max$$ и да се дели на 3

② Вижте в pdf, който съм

(HW) качил задачите:

- DynProgExam.pdf.

Имаме $n > 0$ понички кото във всяко от нея имаме $a_i > 0$ щипци.

Търсим $I = \{i_1, \dots, i_k\} \subseteq \{1, \dots, n\}$ т.ч.

$\sum_{i \in I} a_i \to \max$ и да се дели на 3

$A[1, \dots, n]$

$dp[1, \dots, n][0, 1, 2] \leftrightarrows$

$dp[i][j]$ = max сума от избрани понички от $A[1, \dots, i]$, чиято сума дава j при mod 3

$dp[1][A[1] \% 3] = A[1]$

$dp[1][j] = 0$, $j \in \{0, 1, 2\} \setminus \{A[1] \% 3\}$

for $i \leftarrow 2$ to n do:

  for $j \leftarrow 0$ to 2 do:

    $dp[i][j] \leftarrow \max\{dp[i-1][j],$

    $dp[\underline{i-1}][(j + \underline{A[i]}\%3)\%3] + A[i]\}$

$A[i]$ [illegible]

$\underbrace{\sum_{i \in I} A[i]}, 3$

$C = \left(\sum_{i \in I} A[i]\right) \bmod 3$

$\cancel{\left(\sum_{i \in I} A[i] + b\right)} \underline{(\mathrm{mod}\ 3)} \stackrel{?}{=}$

$\underbrace{\sum_{i \in I} A[i] (\mathrm{mod}\ 3)}_{2} + \underbrace{B \bmod 3}_{2}$

$\underline{dp[n]}[0]$

$T(n) = \Theta(n \cdot 3) = \Theta(n)$, $M(n) = \Theta(n)$

Longest common subsequence.

Дадени са думи $\underline{u}$ и $\underline{v}$. Намерете най-дългата им последователна обща подредица от букви. Нека $|u| = n$, $|v| = m$.

прим: springtime   lcs: printi
printing

The coin change problem

Работите на каса в супермаркет и имате различни типове стотинки n и то в безкрайни количества. Стойността на всяка монета вече я знаете. Как можете да определите по колко начина можете да върнете ресто с тези стотинки при дадено S, което ви подава клиента.

Прим: имате 4 типа стотинки: 8, 3, 1, 2 и съответно за $S=3$ → имаме три начина: $\{1,1,1\}$, $\{1,2\}$ и $\{3\}$. Броят на всички мултимножества A

$A \subseteq_{fin} \mathbf{1} \cup \mathbf{2} \cup \mathbf{3} \cup \mathbf{8}$ т.ч. $\Sigma A = S$

multisubset

$A[1..n]$, $A[i]$ — номинал $A[i]$
все стоимос $A[i]$

1. $A[1..i]$ $\Theta(n \cdot S)$
   $S' \le S$

2. $A[1..i]$ $A[i]$ это добавили?
   $S' \le S$
   - это $A[i] \le S$ тогда
   - пропускаю

3.

```
for i ← 1 to n do
  for S' ← 1 to S do
    dp[i][j] += dp[i-1][S'] + dp[i][S' - A[i]]   if S' ≥ A[i]
    #нолий
```

$dp[0 \ldots n][0, \ldots S]$

$dp[i][j]$ = # начини по които с избор от номинали $A[1], \ldots, A[i]$ можем да конструираме j
(т.е. сумата на избора $= j$)

$dp[0][0] \leftarrow 1$

$dp[i][0] \leftarrow 1$, $1 \leq i \leq n$ // $\underset{\Sigma = 0}{\cancel{\emptyset}} \subseteq A[1] \cup \ldots \cup A[i]$

$dp[0][i] \leftarrow 0$, $1 \leq i \leq S$

$dp[i][j] \leftarrow 0$, $1 \leq i \leq n$ и $1 \leq j \leq S$

```
for i ← 1 to n do
    for s' ← 1 to S do
        if A[i] ≤ s'
            dp[i][s'] += dp[i-1][s'] +
                         dp[i][s' - A[i]]
        else
```

$dp[i][S'] \mathrel{+}= dp[i-1][S']$

Нашият отг. ею $dp[n][S]$.

$T(n,S) = \Theta(n \cdot S)$, $M(n,S) = \Theta(n \cdot S)$

(HW) Помислете как ще стане, ако имаме краен брой монети. Ако искаме пак да намерим минимално по мощност представяне на S като сума от стотинките в задачи?

minStops

Трябва да пътуваме от град A до град B и разстоянието м/у тях е dist.
По пътя има бензиностанции на разстояния $d_1 < d_2 < \ldots < d_n$ от A.
С пълен резервоар колата изминава K километри и е пълен в началото на пътуването.
Предложете алгоритъм, т.е. за време $\Theta(n)$ съставя възможно най-кратък списък от бензиностанции, на които да спрете да зареждате.

Стратегия: ще спираме за зареждане възможно най-късно!

```
minStop(d[1,...,n], dist, k)
    stops[1,...,n], idx ← 1
    lastStop ← 0
    for i ← 2 to n do
        if d[i] - lastStop > k
            stops[idx] ← i-1
            idx ← idx+1
            lastStop ← d[i-1]
    if dist - lastStop > k
        stops[idx] ← n
    return <stops, idx>
```

n-тата бензин.