Pénzváltás probléma

Polonkai Dávid (GPNWZT)

Miskolci Egyetem

2022

Pénzváltás probléma

- Egy pénzösszeget milyen módon tudunk elérni kisebb címletekből?
- Hátizsák feladat speciális esete
- Dinamikus programozással megoldható (részekre bontás)

Hátizsák feladat

Túrázni megyünk. Miket vigyünk a túrára a hátizsákunkba?
Legyen n a lehetséges tárgyak száma.
B a teherbírásunk.
Legyen $a_1, a_2, ..., a_n$ a tárgyak súlya.
Legyen $c_1, c_2, ..., c_n$ a tárgyak értéke (hasznossága)
x_j (0 vagy 1), a j. tárgyat a hátizsákba helyezzük vagy sem.
Feltételek:

$$\sum_{j=1}^{n} c_j x_j \to max$$

$$\sum_{j=1}^{n} a_j x_j \leq B$$

Tehát a nálunk található tárgyak a lehető leghasznosabbak legyenek, de a tömegük ne haladja meg a teherbírásunkat.

Pénzváltás probléma

Egy pénzösszeget akarunk pontosan kifizetni egy adott pénzrendszer címleteivel úgy, hogy minimális számú pénzdarabot használunk fel.

n a pénzrendszer címleteinek a száma.
Legyen $a_1, a_2, ..., a_n$ az egyes címletek értékei.
Legyen B a kifizetendő összeg.
x_j egész szám a j címelet darabszámát mutatja.
Feltételek:

$$\sum_{j=1}^{n} x_j \to min$$

$$\sum_{j=1}^{n} a_j x_j = B$$

Tehát a lehető legkevesebb címlettel fizessük ki a B összeget.

Pénzváltás probléma

Létezik-e megoldás?

$A = \{a_1, a_2, ..., a_n\}$ a címletek halmaza.
b egy pozitív egész szám.
$\sum_{a \in S} = B$, ahol $S \subseteq A$.
Megjegyzés: a címletek csak egyszer használhatóak fel, tehát x_j 0 vagy 1 lehet, és tetszőleges egész számok lehetnek.

Létezik-e megoldás

Egy megoldás:

$$B = a_{i_1} + \cdots + a_{i_k}, i_1 < \ldots < i_k$$

Ekkor

$$B - a_{i_k} = a_{i_1} + \cdots + a_{i_{k-1}}$$

megoldása lesz annak a feladatnak, amelyben a felváltandó érték: $B - a_{i_k}$, felváltásához legfeljebb $(a_1, \ldots, a_{i_{k-1}})$ címleteket használhatjuk.

Részproblémákra bontás

Minden (X, i) $(1 \leq X \leq B, 1 \leq i \leq n)$számpárra, X felváltható-e az első $a_1, ..., a_i$ pénzzel.
$V(X, i) = \mathit{Igaz}$, ha az első i pénzzel előállítható.
$V(X, i) = \mathit{Hamis}$, ha az első i pénzzel nem állíható elő.

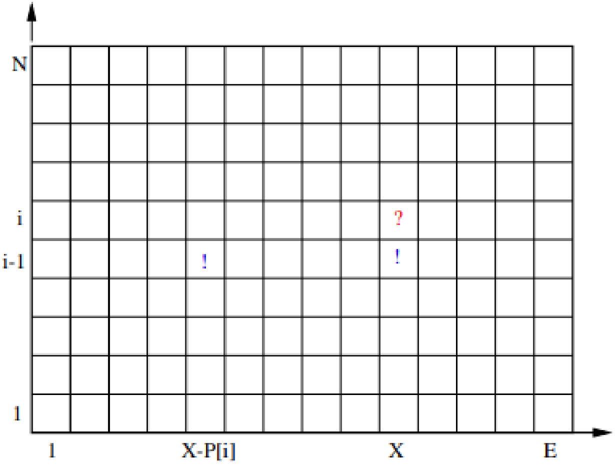
Részproblémák eredménye (Rekurzív megoldás)

$$V(X,i) \Leftrightarrow \begin{cases} X = a_i \vee \\ i > 1 \wedge V(X, i-1) \vee \\ i > 1 \wedge > a_i \wedge V(X - a_i, i-1) \end{cases}$$

Részproblémák eredménye (Táblázatos módszer)

Kiszámítási sorrend módosítása, hogy először azokat számoljuk ki, amelyek szükségesek a későbbi számításokhoz.
(X, i) probléma összetevői: $(X, i-1)$ és $(X - a_i, i-1)$
Ezért a következő táblázatot soronként alulról felfele, balról jobbra töltjük ki.

Táblázat



1. ábra. A pénzváltás táblázata

A Pénzváltás probléma, egy felváltás eloállítása

Megoldás csak akkor létezhet, ha az aktuális problémára nézve $V[B, n] = Igaz$.
Keressük azt a legkisebb i értéket, amelyre $V[B, i] = Igaz$, tehát $V[B, i - 1] = Hamis$.
Ebből tujuk, hogy a_i pénz szerepel a B-ben.
Folytatva ezt a folyamatot $B - a_i$-re és $i - 1$-re,
azaz $V[B - a_i, i - 1]$-re addig, amíg $B = 0$-t nem kapunk.

Optimális pénzváltás

Az előző diában bemutatott mohó stratégia nem optimális, mivel 8 = 5+1+1+1 eredményt adna.
Az optimális megoldáshoz a már előzőleg bevezetett részproblémákra bontást alkalmazzuk.
Minden (X, i) $(1 \leq X \leq B, 1 \leq i \leq n)$ számpárra, legkevesebb hány pénz összegeként lehet az X-et előállítani legfeljebb az első $i\{a_1, \ldots, a_i\}$ pénz felhasználásával. Ha nincs megoldás, akkor legyen $n + 1$.
A probléma optimális megoldása: $Opt(X, i)$ Az optimális megoldás értéke: $X = 0$-ra és $i = 0$-ra, $Opt(X, 0) = n + 1$ és $Opt(0, i) = 0$.
Mindezek alapján az alábbi rekurzív összefüggés írható fel

Optimális pénzváltás

$$Opt(X,i) = \begin{cases} \infty & \text{ha } i = 0 \wedge X > 0 \\ 0 & \text{ha} X = 0 \\ Opt(X, i-1) & \text{ha} X < a_i \\ min(Opt(X, i-1), & \\ 1 + Opt(X - a_i, i-1)) & \text{ha} X \geq a_i \end{cases}$$

Optimális pénzváltás

Az előző rekurzív kifejezést, színtén táblázatba tudjuk foglalni, mivel $Opt(X, i)$-nek legfeljebb $Opt(X - a_i, i - 1)$-re vagy $Opt(X, i - 1)$-re van szüksége.
A táblázatból a felváltás előállítását, a már bemutatott módszer módosításával hajthatjuk végre.

Optimális pénzváltás, egy optimális felváltás előállítása

Megoldás csak akkor létezhet, ha az aktuális problémára nézve $V[B, n] < \infty$.
Keressük azt a legkisebb i értéket, amelyre $V[B, i] \to min$, tehát $V[B, i-1] > V[B, i]$ (hiszen $a_i > a_{i-1}$).
Ebből tujuk, hogy a_i pénz szerepel a B optimális felváltásában.
Folytatva ezt a folyamatot $B - a_i$-re és $i - 1$-re,
azaz $V[B - a_i, i - 1]$-re addig, amíg $B = 0$-t nem kapunk.

Végtelen aprópénz

Egy pénzösszeget akarunk pontosan kifizetni egy adott pénzrendszer címleteivel úgy, hogy minimális számú pénzdarabot használunk fel és az egyes címletekből korlátlan mennyiség áll rendelkezésre

n a pénzrendszer címleteinek a száma.

Legyen $a_1, a_2, ..., a_n$ az egyes címletek értékei.

Legyen B a kifizetendő összeg.

x_j egész szám a j címelet darabszámát mutatja.

Feltételek:

$$\sum_{j=1}^{n} x_j \rightarrow min$$

$$\sum_{j=1}^{n} a_j x_j = B$$

Tehát a lehető legkevesebb címlettel fizessük ki a B összeget.

Végtelen aprópénz

$$E(0) = 0$$

$$E(B) = 1 + min \begin{cases} E(B - a_1) \text{ ahol } B - a_1 >= 0 \\ E(B - a_2) \text{ ahol } B - a_2 >= 0 \\ \dots \\ E(B - a_n) \text{ ahol } B - a_n >= 0 \\ \text{Ha mindegyik esetben } B - a_x < 0, \\ \text{akkor nem bontható fel} \end{cases}$$

Ahol $E(x)$, az x érték felbontásához szükséges érmék számát tartalmazza. Mindezekből a rekurzív megoldás egyértelmű.

Végtelen aprópénz dinamikus programozás

Ha az előzőleg használt módszert használjuk, azaz eltároljuk a kiszámított megoldásokat, és $E(1)$-től $E(B)$-ig haladunk, megkapjuk a dinamikus programozott megoldást.

Felhasznált irodalmak

1. Horváth Gyula Szegedi Tudományegyetem Természettudományi és Informatikai Kar Algoritmizálás jegyzet
2. Dr. Házy Attila: Nemlineáris optimalizálás. (elektronikus jegyzet)
3. J. W. Wright. 1975. The Change-Making Problem. J. ACM 22, 1 (Jan. 1975), 125–128. https://doi.org/10.1145/321864.321874
4. (BME Programozási stratégiák órai feladatok)

KÖSZÖNÖM A FIGYELMET!

https://davidpolonkai.github.io/change_making_problem/