Nem kifizethető címlet

Innen: Algowiki

Tartalomjegyzék

Feladat

N (1≤N≤500) különböző természetes szám értékből határozzuk meg, hogy mely 1 és M közötti számot nem lehet előállítani a megadott N értékek összegeként.

- 1≤N≤500
- 1≤M≤100000

Az eredeti feladat

mester.inf.elte.hu

- Téma:
  - Szint: Haladó
  - Téma: Dinamikus programozás
- Feladat: 69. Nem kifizethető címletek

Megoldás

Fontos gondolatok

Mindegyik számról tudjuk, hogy elő lehet állítani azokból a számokból amik a megadott értékekkel kisebbek nála.

Pl.: 2,3 a két érték -> az 5 előállítható 5-2 = 3-ból és 5-3 = 2-ből. Ha azok a számok is előállíthatóak amikből ez előállítható, akkor a kiindulásból is előállítható lesz.

A kiindulás a 0, mert az elállítható ha minden értékből nullát adunk össze.

Részletes megoldás rekurzióval

A gondolatból kiindulva meg is van a rekurzív megoldás.

A függvényünkben minden számnál végigmegyünk az értékeken és meghívjuk a függvényt a számnál az adott értékkel kisebb számra.

Ha bármelyik igazzal tér vissza, akkor ez a szám is igazat ad vissza, különben hamisat.

A rekurzió végei a nulla, ami alapból igazat ad vissza, és a negatív számok, amik hamisat adnak vissza.

Optimalizálás ha csak az első igazat visszaadó részhívásig megy.

Dinamikus programozás ötletével eltároljuk a már eddig kiszámolt adatokat és utána azokkal dolgozunk.

Így ha olyanra hívjuk meg a rekurziót ami már megvan, akkor csak visszaadjuk az értékét.

Komplexitás

A beolvasásnál N elemet eltárolunk. A kiírásnál M elemet ellenőrzünk és kiírunk.

A feladat megoldásánál M elemre hívjuk meg a függvényt, ami még N elemre hívja meg, amik még N elemre hívják meg...

Ez megy a rekurzió végéig. Minél kisebbek az értékek annál mélyebb lesz a rekurzió.

Az *optimalizálásokkal* tudjuk, hogy minden számra a nála kisebbek előállíthatóságát már kiszámoltuk. Így a belső rekurzív függvényei konstans időben visszaadják az eredményüket.

- Beolvasás: Θ(N)
- Kiírás: Θ(M)
- *Optimalizált* megoldás: O(M*N)

Implementáció

C++ implementáció (https://pastebin.com/dCzjhGJT)

Példa

N:2 M:7

Értékek: 2 3

Számok	1	2	3	4	5	6
Rekurzió	1-2= -1-> Hamis	2-2= 0-> Igaz	3-2= 1->Hamis	4-2= 2->Igaz	5-2= 3->Igaz	6-2= 4->Igaz
	1-3= -2-> Hamis	Ide már nem fut	3-3= 0->Igaz	Ide már nem fut	Ide már nem fut	Ide már nem fut
Eredmény	Hamis	Igaz	Igaz	Igaz	Igaz	Igaz

Részletes megoldás sorral

Az ötlet alapján sorokat felhasználva is megoldhatjuk a feladatot.

A sor első elemének az értékekkel növeltjeit berakjuk a sor végére, aztán kivesszük az elsőt. A kiinduló elemmel kezdve, az összes számról ami volt a sorban tudjuk, hogy előállítható lesz.

Optimalizálás ha egy már kiszámolt szám jön a sorban akkor azt csak kiszedjük. Nem teszünk be semmit. Így itt is minden számot csak egyszer számolunk ki.

Komplexitás

A beolvasás és kiírás nem változik.

Nullából kiindulva berakunk N számot. Mindegyik számra berakunk N számot. Ez N alapú exponenciális műveletigény lenne.

Optimalizálva a legrosszabb esetben minden M szám előállítható. Ekkor mindegyik szám berakja az általa előállítható N számot. De minden szám csak egyszer.

- Beolvasás: $\Theta(N)$
- Kiírás: $\Theta(M)$
- *Optimalizált* megoldás: $O(M*N)$

Implementáció

C++ implementáció (https://pastebin.com/qfDsKnr5)

Példa

N:2 M:10

Értékek: 2 3

Sor

- { 0 } -> { 2,3 }
- { 2,3 } -> { 4,5,5,6 }
- { 4,5,5,6 } -> { 6,7,7,8,(7,8),8,9 }

A zárójelben lévők az *optimalizált* verzióban nem kerülnek bele a sorba mert a második ötössel már nem számolunk.

- { 6,7,7,8,8,9 } -> { 8,9,9,10,(9,10),10,[11],(10,11),[11],[12] }

A szögletes zárójelben lévők már a tartományon kívül esnek.

Részletes megoldás ciklussal

A legegyszerűbb dinamikus programozási megoldás egy ciklus ami minden számra végigmegy az értékeken és beállítja a szám értékkel megnöveltjét úgy, hogy az előállíthatóságát vagyoljuk a száméval.

Optimalizálás ha csak az előállítható számokon dolgozunk.

Komplexitás

A beolvasás és kiírás nem változik.

Ebben a megoldásban is M számon kell bejárni N értéket.

- Beolvasás: Θ(N)
- Kiírás: Θ(M)
- *Optimalizált* megoldás: O(M*N)

Implementáció

C++ implementáció (https://pastebin.com/aJaDXGPF)

Példa

N:2 M:7

Értékek: 2 3

A zárójelben lévőt az *optimalizáltban* nem állítjuk be.

Kiindulás	0=Igaz	1=Hamis	2=Hamis	3=Hamis	4=Hamis	5=Hamis	6=Hamis	7=Hamis
0.	0=Igaz	1	2=Igaz	3=Igaz	4	5	6	7
1.	0	1=Hamis	2	3=Igaz(v Hamis)	4(=Hamis)	5	6	7
2.	0	1	2=Igaz	3	4=Igaz	5=Igaz	6	7
3.	0	1	2	3=Igaz	4	5=Igaz	6=Igaz	7
4.	0	1	2	3	4=Igaz	5	6=Igaz	7=Igaz
5.	0	1	2	3	4	5=Igaz	6	7=Igaz
6.	0	1	2	3	4	5	6=Igaz	7
7.	0	1	2	3	4	5	6	7=Igaz

A lap eredeti címe: „https://algowiki.miraheze.org/w/index.php?title=Nem_kifizethető_címlet&oldid=1353”