1. Gyakorlat

F1. Vegyük a következő tükörszavas problémát. Határozzuk meg minimálisan hány betűt kell beszúrni egy szóba, hogy tükörszó legyen.

Megold: A feladat a törlés esetéhez nagyon hasonlóan megoldható. Legyen $T[i,j]$ azon részprobléma megoldása, amelyben tükörszóvá kell tenni az i-edik karaktertől a j-edik karakterig tartó szót. Ekkor $T[i,i] = 0$, hiszen az egy betűből álló szó az tükörszó. Továbbá $T[i,i+1] = 0$, ha az i-edik és $i+1$-edik betűk megegyeznek, és $T[i,i+1] = 1$ egyébként. A további értékekre a függvényérték rekurzívan számolható. Ha az i-edik karakter megegyezik a j-edikkel, akkor $T[i,j] = T[i+1,j-1]$. Amennyiben nem egyeznek meg, akkor valamelyik szélső karaktert fel kell venni a másik oldalra és ezzel a beszúrással egy további részproblémára vezetjük vissza a feladatot. Ebben az esetben két lehetőség közül választjuk a jobbikat, így ekkor

$$T[i,j] = \min\{T[i+1,j]+1, T[i,j-1]+1\}$$

A rekurzió alapján kiszámolhatóak a $T[i,j]$ értékek, és a megoldás a $T[1,n]$ függvényérték. Arra kell ügyelnünk, hogy mindig egy már megoldott problémára vezessük vissza a feladatot, ez teljesül, ha az értékeket növekvő $j-i$ érték alapján számoljuk ki (pontosan úgy, mint a törlés esetén).

F2. Adott egy $k \times n$-es tábla. Minden mezőre meg van adva egy c_{ij} pozitív szám, ami a mezőről begyűjthető érték. Egy játékos a bal alsó sarokból szeretne eljutni a jobb felső sarokba úgy, hogy csak jobbra és felfelé léphet szomszédos mezőre. Az útja során, összegyűjtheti a mezőkről az értékeket. Mi az az útvonal, amivel a maximális értéket tudja összegyűjteni.

Megold: Legyen $F(x,y)$ a maximális érték, amit az (x,y) mezőig össze tudunk gyűjteni. Ekkor a kezdeti értékek $F(1,y) = \sum_{j=1}^{y} c_{1j}$ és $F(x,1) = \sum_{i=1}^{x} c_{i1}$, hiszen ezekben az esetekben csak felfele, illetve csak jobbra mehet a játékos. A közbenső mezőkre alulról vagy felülről léphetünk, így

$$F(x,y) = \max\{F(x-1,y) + c_{xy}, F(x,y-1) + c_{x,y}\}$$

Kiszámolva ezeket az értékeket az $F(k,n)$ érték adja meg, hogy mit gyűjthetünk össze maximálisan.

Az optimális megoldást visszafejtéssel megtalálhatjuk, ha minden $F(x,y)$ érték esetén megjegyezzük, melyik választás adta a rekurzióban a maximumot.

F3. Vegyük a következő ütemezési problémát. Adott három gép és munkák, minden munkának van egy p_i végrehajtási ideje. Feladatunk a munkákat a gépekhez rendelni úgy, hogy a gépekhez rendelt munkák végrehajtási idejeinek összegének (mind a három gépre kiszámolva ezt az összeget) a maximuma minimális legyen.

Megold A dinamikus programozási eljáráshoz jelölje $F_i(x,y)$ $(x,y \leq \sum p_i)$ azt a minimális összeget, amelyet elérhetünk az első i munka ütemezése után a harmadik gépen ha az első két gépen az összeg legfeljebb x és y.

Az $F_1(x, y)$ értéke 0, ha az első munka kisebb, mint $\max\{x, y\}$ és p_1 egyébként.

A további értékek az $F_{i+1}(x, y) = \min\{F_i(x-p_{i+1}, y), F_i(x, y-p_{i+1}), F_i(x, y)+p_{i+1}\}$ rekurzióval határozhatóak meg. A rekurziónál használt szétválasztás alapja az, hogy az $i+1$-edik munkát melyik gépre ütemezzük, ha az első gépre az első értékhez jutunk (ami egy már megoldott probléma optimuma), ha a másodikra a másodikhoz, ha a harmadikra a harmadikhoz. Így ezek minimuma lesz az F_{i+1} értéke.

Az optimális megoldás célfüggvényértékét az utolsó F_n függvény alapján találhatjuk meg, venni kell minden x, y-ra a $\max\{x, y, F_n(x, y)\}$ értéket és ezek minimuma lesz a célfüggvényérték.

Az optimális megoldást visszafejtéssel megtalálhatjuk, ha minden $F_i(x, y)$ érték esetén megjegyezzük, melyik választás adta a rekurzióban a minimumot.

Megjegyzés: A módszer általánosítható, több gép esetére, illetve más célfüggvényekre is.