



Feladatmegoldási stratégiák Visszalépéses keresés II.



Feladatmegoldási stratégiák

Visszalépéses keresés



Feladat – 1. változat

Egy vállalkozás N különböző állásra keres munkásokat. Pontosan N jelentkező érkezett, ahol minden jelentkező megmondta, hogy mely munkákhoz ért, illetve amihez ért.

A vállalkozás vezetője azt szeretné, ha az összes jelentkezőt fel tudná venni és minden munkát elvégeztetni.

$M(i)$ – az i . munkás ennyi munkához ért

$E(i, j)$ – az i . munkás által elvégezhető j . munka

Darab Állások: 1. 2. 3.

1. jelentkező:	2	1	4	
2. jelentkező:	1	2		
3. jelentkező:	2	1	2	
4. jelentkező:	1	3		
5. jelentkező:	3	1	3	5





Feladatmegoldási stratégiák

Visszalépéses keresés



N munka – N jelentkező:

Keresés (N, Van, Y) :

$i := 1; Y := (0, \dots, 0)$

Ciklus amíg $i \geq 1$ és $i \leq N$ {lehet még és nincs még kész}

Jóesetkeresés (i, Van, j)

Ha Van akkor $Y(i) := j; i := i + 1$ {előrelépés}

különben $Y(i) := 0; i := i - 1$ {visszalépés}

Ciklus vége

$Van := (i > N)$

Eljárás vége.





Feladatmegoldási stratégiák

Visszalépéses keresés



N munka – **N** jelentkező:

Jóesetkeresés (i, Van, j) :

$j := Y(i) + 1$

Ciklus amíg $j \leq M(i)$ és rossz(i, j)

$j := j + 1$

Ciklus vége

$\text{Van} := (j \leq M(i))$

Eljárás vége.





Feladatmegoldási stratégiák

Visszalépéses keresés



N munka – **N** jelentkező:

`rossz(i, j) :`

`k:=1`

Ciklus amíg $k < i$ és $E(k, Y(k)) \neq E(i, j)$

`k:=k+1`

Ciklus vége

`rossz := (k < i)`

Eljárás vége.

Darab Állások: 1. 2. 3.

1. jelentkező:	2	1	4	
2. jelentkező:	1	2		
3. jelentkező:	2	1	2	
4. jelentkező:	1	3		
5. jelentkező:	3	1	3	5

$E(i, j)$ – az i . munkás által elvégezhető
 j . munka





Feladatmegoldási stratégiák

Visszalépéses keresés



Feladat – 2. változat

Egy vállalkozás N különböző állásra keres munkásokat. Pontosan N jelentkező érkezett, ahol minden jelentkező megmondta, hogy mely munkákhoz ért, illetve amihez ért.

A vállalkozás vezetője azt szeretné, ha az összes jelentkezőt fel tudná venni és minden munkát elvégeztetni.

$F(i, j)$ – az i . munkás ért-e a j . munkához?

i	h	h	i	h
h	i	h	h	h
i	i	h	h	h
h	h	i	h	h
i	h	i	h	i





Feladatmegoldási stratégiák

Visszalépéses keresés



N munka – N jelentkező:

Keresés (N, Van, Y) :

$i := 1; Y := (0, \dots, 0)$

Ciklus amíg $i \geq 1$ és $i \leq N$ {lehet még és nincs még kész}

Jóesetkeresés (i, Van, j)

Ha Van akkor $Y(i) := j; i := i + 1$ {előrelépés}

különben $Y(i) := 0; i := i - 1$ {visszalépés}

Ciklus vége

$Van := (i > N)$

Eljárás vége.





Feladatmegoldási stratégiák

Visszalépéses keresés



N munka – **N** jelentkező:

Jóesetkeresés (i, Van, j):

$j := Y(i) + 1$

Ciklus amíg $j \leq N$ és

(rossz(i, j) vagy nem $F(i, j)$)

$j := j + 1$

Ciklus vége

$\text{Van} := (j \leq N)$

Eljárás vége.

i	h	h	i	h
h	i	h	h	h
i	i	h	h	h
h	h	i	h	h
i	h	i	h	i





Feladatmegoldási stratégiák

Visszalépéses keresés



N munka – **N** jelentkező:

`rossz(i, j) :`

`k:=1`

Ciklus amíg $k < i$ és $Y(k) \neq j$

`k:=k+1`

Ciklus vége

`rossz := (k < i)`

Eljárás vége.

i	h	h	i	h
h	i	h	h	h
i	i	h	h	h
h	h	i	h	h
i	h	i	h	i





Feladatmegoldási stratégiák

Visszalépéses keresés



Feladat

Egy pályaválasztási intézet elhatározza, hogy a 8. osztályos tanulók iskolaválasztásai alapján (minden jelentkezési lapon maximum két iskolát lehet megjelölni) megpróbál olyan 'beiskolázást' megvalósítani, amelyben minden tanulót az általa megjelölt valamelyik iskolába fel is vesznek. (Tudjuk az egyes iskolákba felvehetőek számát.) Adj meg egy lehetséges jó beiskolázást!





Feladatmegoldási stratégiák

Visszalépéses keresés



N tanuló beiskolázása **M** iskolába:

Keresés (N, Van, Y) :

$i := 1; Y := (0, \dots, 0)$

Ciklus amíg $i \geq 1$ és $i \leq N$ {lehet még és nincs még kész}

Jóesetkeresés (i, Van, j)

Ha Van akkor $Y(i) := j; i := i + 1$ {előrelépés}

különben $Y(i) := 0; i := i - 1$ {visszalépés}

Ciklus vége

$Van := (i > N)$

Eljárás vége.





Feladatmegoldási stratégiák

Visszalépéses keresés



N tanuló beiskolázása **M** iskolába:

Jóesetkeresés (i, Van, j) :

$j := Y(i) + 1$

Ha $Igény(i, 2) = 0$ akkor $K := 1$ különben $K := 2$

Ciklus amíg $j \leq K$ és rossz (i, j)

$j := j + 1$

Ciklus vége

$\text{Van} := (j \leq K)$

Eljárás vége.





Feladatmegoldási stratégiák

Visszalépéses keresés



N tanuló beiskolázása **M** iskolába:

`rossz(i, j) :`

`db := 1`

Ciklus $k=1$ -től $i-1$ -ig

Ha $Igény(k, Y(k)) = Igény(i, j)$ akkor `db := db + 1`

Ciklus vége

`rossz := (db > Kapacitás(Igény(i, j)))`

Eljárás vége.





Feladatmegoldási stratégiák

Visszalépéses keresés



További visszalépéses keresés feladatok:

- Labirintusban útkeresés
- Permutációk, kombinációk előállítása
- Térképszínezés
- Pénzfelbontás adott címletekre





Feladatmegoldási stratégiák

Visszalépéses keresés



Visszalépéses keresés feladatok: (végtelen eset)

- Úthossz-korlát: Fává egyenesítünk, végtelen fát állítunk elő. Nem engedjük, hogy az aktuális út hossza meghaladja az úthossz-korlátot.
 - Ha túl rövidre választjuk az úthossz-korlátot (túl alacsonyan vágjuk el a gráfot) akkor nem találunk megoldást.
 - Ha a start csúcsban áll elő a visszalépési feltétel, akkor:
 - 1. nincs megoldás
 - 2. túl rövidre választottuk az úthossz-korlátot





Feladatmegoldási stratégiák

Visszalépéses keresés



Visszalépéses keresés feladatok:

- Kör kiküszöbölése
 - lesz egy újabb visszalépési feltétel: az aktuális csúcs szerepelt-e már az aktuális úton
 - ha igen: rögtön visszalépés (így nem zárjuk be a kört).





Feladatmegoldási stratégiák

Visszalépéses keresés



A visszalépéses stratégia

- véges fákban mindig terminál (véges fákban teljes);
- végtelen gráfban úthossz-korláttal terminál (kör kizárása: az aktuális út csúcsait nem engedjük ismételni);
- egy zsákutcát többször is bejár, ha több út vezet hozzá.





Feladatmegoldási stratégiák

Visszalépéses kiválogatás



Visszalépéses kiválogatás rekurzív algoritmus:

Visszalépéses kiválogatás (N, Db, Y) :

$Db := 0$; $X := (0, \dots, 0)$; Backtrack $(1, N, X, Db, Y)$

Eljárás vége.





Feladatmegoldási stratégiák

Visszalépéses kiválogatás



Backtrack(i, N, X, Db, Y):

Ha $i=N+1$ akkor $Db:=Db+1$; $Y(Db):=X$

különben Ciklus $j=1$ -től $N-i$ ig

Ha nem tilos(i, j) és nem Rossz(i, j)
akkor $X(i):=j$

Backtrack($i+1, N, X, Db, Y$)

Ciklus vége

Elágazás vége

Eljárás vége.





Feladatmegoldási stratégiák

Visszalépéses maximumkeresés



Visszalépéses maximumkeresés rekurzív algoritmus:

Visszalépéses maximumkeresés (N, Van, Y) :

$X := (0, \dots, 0)$; $Y := X$; Backtrack $(1, N, X, Y)$

$Van := Y \neq (0, \dots, 0)$

Eljárás vége.





Feladatmegoldási stratégiák

Visszalépéses maximumkeresés



Visszalépéses maximumkeresés rekurzív algoritmus:

Backtrack(i, N, X, Y):

Ha $i=N+1$ akkor **ha nagyobb? (X, Y) akkor $Y:=X$**

különben Ciklus $j=1$ -től N -ig

Ha nem tilos(i, j) és nem Rossz(i, j)
akkor $X(i):=j$

Backtrack($i+1, N, X, Y$)

Ciklus vége

Elágazás vége

Eljárás vége.





Feladatmegoldási stratégiák

Visszalépéses maximumkeresés



Példa: (1. változat)

Egy vállalkozás N különböző állásra keres munkásokat. Pontosan N jelentkező érkezett, ahol minden jelentkező megmondta, hogy mely munkákhoz ért, illetve amihez ért, arra mennyi fizetést kérne.

Minden munkát el kell végeztetni valakivel, mindenkinek munkát kell adni, de a legolcsóbban!

Allások: 1. 2. 3. 4. 5.

1. jelentkező:	100	0	0	100	0
2. jelentkező:	0	200	0	0	0
3. jelentkező:	200	100	0	0	0
4. jelentkező:	0	0	200	0	400
5. jelentkező:	500	0	400	0	200





Feladatmegoldási stratégiák

Visszalépéses maximumkeresés



Ha egy megoldás elkészül, akkor a költségét így számíthatjuk ki:

költség (X) :

$S := 0$

Ciklus $i=1$ -től N -ig

$S := S + F(i, X(i))$

Ciklus vége

Függvény vége.

Kezdetben olyan – fiktív – megoldásból kell kiindulni, aminél minden valódi megoldás jobb.





Feladatmegoldási stratégiák

Visszalépéses maximumkeresés



Legjobb állás (N, i) :

Ha $i > N$ akkor

Ha költség $(X) < \text{költség}(Y)$ akkor $Y := X$

különben

Ciklus $j=1$ -től $N-i$ -ig

Ha nem volt (i, j) és $F(i, j) > 0$

akkor $X(i) := j$; Legjobb állás $(N, i+1)$

Ciklus vége

Elágazás vége

Eljárás vége.

Ebben a megoldásban feleslegesen sokszor hívjuk a Költség függvényt.





Feladatmegoldási stratégiák

Visszalépéses maximumkeresés



Legjobb állás (N, i) :

Ha $i > N$ akkor $költ := költség(X)$

Ha $költ < legjobb$

akkor $Y := X; legjobb := költ$

különben

Ciklus $j=1$ -től N -ig

Ha nem volt (i, j) és $F(i, j) > 0$

akkor $X(i) := j; legjobb állás(N, i+1)$

Ciklus vége

Elágazás vége

Eljárás vége.



Itt feleslegesen nem hívjuk a Költség függvényt, jó legjobb kezdőérték kell.



Feladatmegoldási stratégiák

Visszalépéses maximumkeresés



Már csak egy apróságra gondolhatunk: ha van egy megoldásunk és a most készülő megoldásról látszik, hogy már biztosan rosszabb lesz – többbe fog kerülni –, akkor azt már nem érdemes tovább vinni.

Legyen az eljárás paramétere az eddigi költség, s az eljárást csak akkor folytassuk, ha még nem érjük el a korábban kiszámolt maximális költséget. Emiatt nem a megoldások elkészültekor kell számolni költséget, hanem menet közben, folyamatosan.





Feladatmegoldási stratégiák

Visszalépéses maximumkeresés



Legjobb állás $(N, i, \text{költ})$:

Ha $i > N$ akkor

Ha $\text{költ} < \text{legjobb}$ akkor $Y := X$; $\text{legjobb} := \text{költ}$

különben Ciklus $j=1$ -től N -ig

Ha nem volt (i, j) és $F(i, j) > 0$

és $\text{költ} + F(i, j) < \text{legjobb}$

akkor $X(i) := j$

Legjobb állás $(N, i+1, \text{költ} + F(i, j))$

Ciklus vége

Elágazás vége

Eljárás vége.





Feladatmegoldási stratégiák

Visszalépéses maximumkeresés



Példa: (2. változat)

Egy vállalkozás N különböző állásra keres munkásokat. Pontosan M jelentkező érkezett ($M < N$), ahol minden jelentkező megmondta, hogy mely munkákhoz ért, illetve amihez ért, arra mennyi fizetést kérne.

Mindenkinek munkát kell adni (csak egyet mindenkinek), de a legolcsóbban!

Állások: 1. 2. 3. 4. 5.

1. jelentkező:	100	0	0	100	0
2. jelentkező:	0	200	0	0	0
3. jelentkező:	200	100	0	0	0
4. jelentkező:	0	0	200	0	400





Feladatmegoldási stratégiák

Visszalépéses maximumkeresés



Legjobb állás $(N, M, i, \text{költ})$:

Ha $i > M$ akkor

Ha $\text{költ} < \text{legjobb}$ akkor $Y := X$; $\text{legjobb} := \text{költ}$

különben Ciklus $j=1$ -től N -ig

Ha nem volt (i, j) és $F(i, j) > 0$

és $\text{költ} + F(i, j) < \text{legjobb}$

akkor $X(i) := j$

Legjobb állás $(N, M, i+1, \text{költ} + F(i, j))$

Ciklus vége

Elágazás vége

Eljárás vége.





Feladatmegoldási stratégiák

Visszalépéses maximumkeresés



Példa: (3. változat)

Egy vállalkozás N különböző állásra keres munkásokat. Pontosan M jelentkező érkezett ($M > N$), ahol minden jelentkező megmondta, hogy mely munkákhoz ért, illetve amihez ért, arra mennyi fizetést kérne.

Mindenkinek munkát el kell végezni, egy-egy embernek, de a legolcsóbban!

Állások: 1. 2. 3. 4.

1. jelentkező:	100	0	0	100
2. jelentkező:	0	200	0	0
3. jelentkező:	200	100	0	0
4. jelentkező:	0	0	200	0
5. jelentkező:	500	0	400	0





Feladatmegoldási stratégiák

Visszalépéses maximumkeresés



Legjobb állás $(N, M, i, \text{költ})$:

Ha $i > N$ akkor

Ha $\text{költ} < \text{legjobb}$ akkor $Y := X$; $\text{legjobb} := \text{költ}$

különben Ciklus $j=1$ -től M -ig

Ha nem volt (j, i) és $F(j, i) > 0$

és $\text{költ} + F(j, i) < \text{legjobb}$

akkor $X(i) := j$

Legjobb állás $(N, M, i+1, \text{költ} + F(j, i))$

Ciklus vége

Elágazás vége

Eljárás vége.





Feladatmegoldási stratégiák Visszalépéses keresés II.