Mohó stratégia

Mohó stratégia

Optimalizálási probléma megoldására szolgáló algoritmus gyakran olyan lépések sorozatából áll, ahol minden lépésben adott halmazból választhatunk.

A mohó algoritmus mindig az adott lépésben optimálisnak látszó választást teszi, a lokális optimumot választja abban a reményben, hogy ez globális optimumhoz fog majd vezetni.

A mohó algoritmus nem mindig ad optimális megoldást, azonban sok probléma megoldható mohó algoritmussal.

Az előadás Horváth Gyula tananyagai felhasználásával készült.

Mohó stratégia

Feladat:

N-féle pénzjegyünk van, P_1, P_2, …, P_n címletű ($P_i<P_{i+1}$). Add meg, hogy minimálisan melyek felhasználásával fizethető ki az F összeg! Feltehetjük, hogy minden pénzjegyből tetszőleges számú van.

Megoldás:

Vegyünk a legnagyobb címletű pénzjegyből, amennyi szüksé-ges, majd a maradék összeget fizessük ki a nála kisebb pénz-jegyekkel!

Mohó stratégia

```
Pénzváltás(N,P,F,Db):
  i:=N
  Ciklus amíg i>0 és F>0
    db(i):=F div P(i); F:=F mod P(i)
    i:=i-1
  Ciklus vége
  Ha i=0 akkor ...
Eljárás vége.
```

Probléma: P=(1,3,4), F=6 esetén a megoldás (2,0,1), azaz 3 pénzjeggyel fizetnénk ki a 6 forintot, pedig 6=3+3!

A helyes működés feltétele: $2*P(i) \leq P(i+1)$.

Mohó stratégia

Hátizsák probléma

N-féle anyagot kell egy hátizsákba pakolni (darabolni is lehet), V_i az anyag értéke, W_i a súlya, a hátizsákba H súly fér, a lehető legnagyobb értéket kell elvinni.

Rendezzük az anyagokat V_i/W_i szerint! Lássuk be, hogy e sorrend szerint folyamatosan kell vennünk, amíg lehet, esetleg az utolsónak csak egy részét!

Mohó stratégia

Bizonyítás

Tegyük fel, hogy az első anyagból 1 kg-mal kevesebbet te-szünk a hátizsákba! Ekkor az össz-érték V_1/W_1 –gyel csökken.

Ha bármely későbbiből veszünk 1 kg-ot, akkor az össz-érték V_i/W_i –vel nő, de mivel $V_1/W_1 \geq V_i/W_i$, ezért az össz-érték így nem lehet nagyobb!

Mohó stratégia

```
Hátizsák(N,V,W,H):
  Rendezés(N,V,W,S)
  Összeg:=0; i:=1
  Ciklus amíg i≤N és Összeg+W(i)≤H
    Ki: S(i)-ből W(i)
    Összeg:=Összeg+W(i); i:=i+1
  Ciklus vége
  Ha i≤N akkor
     Ha Összeg<H akkor Ki: S(i)-ből H-Összeg
Eljárás vége.
```

Kérdés: érdemes előre kiszámolni, hogy mind el tudjuk-e vinni?

Mohó stratégia

Hátizsák probléma

Ha az egyes anyagok nem darabolhatók, akkor a feladat mohó stratégiával nem oldható meg. Ellenpélda a mohó stratégiára:

$W_1=10$, $V_1=60$, $V_1/W_1=6$

$W_2=20$, $V_2=100$, $V_2/W_2=5$

$W_3=30$, $V_3=120$, $V_3/W_3=4$

Hátizsák kapacitás: 50

Ekkor a mohó megoldás: (1 és 2) 160, pedig a jó megoldás (2 és 3) 220.

Az sem jó, ha a legkönnyebbet vagy a legtöbbet érőt választjuk.

Mohó stratégia

Feladat: (Kása Zoltántól)

Egy ügyintézőnek egy napon N ügyféllel kell foglalkoznia, tetszőleges sorrendben. Tudjuk előre, hogy melyik ügyet hány perc alatt tudja megoldani. Adj meg egy olyan ügyfélsorrendet, amelyben a várakozási idők összege a lehető legkisebb legyen!

Megoldás:

Rendezzük sorba az ügyeket elintézési idő szerint növekvő sorrendbe, majd vegyük sorra az ügyfeleket!

Mohó stratégia

Egy konkrét példa:

$n = 3\ t_1 = 60,\ t_2 = 10,\ t_3 = 30.$

sorrend	összvárakozási idő
1,2,3	0+60+(60+10)=130
1,3,2	0+60+(60+30)=150
2,3,1	0+10+(10+30)= 50
2,1,3	0 +10+(10+60)=80
3,1,2	0+30+(30+60)=120
3,2,1	0+30+(30+10)=70

Várakozási idő=
(n-1)*t(1)+(n-2)*t(2)+
...
+2*t(n-2)+1*t(n-1)

Mohó stratégia

Feladat:

Egy kábelhálózat különböző csatornáin N filmet játszanak. Ismerjük mindegyik film kezdési és végidejét. Egyszerre csak 1 filmet tudunk nézni. Add meg, hogy maximum hány filmet nézhetünk végig!

Megoldás:

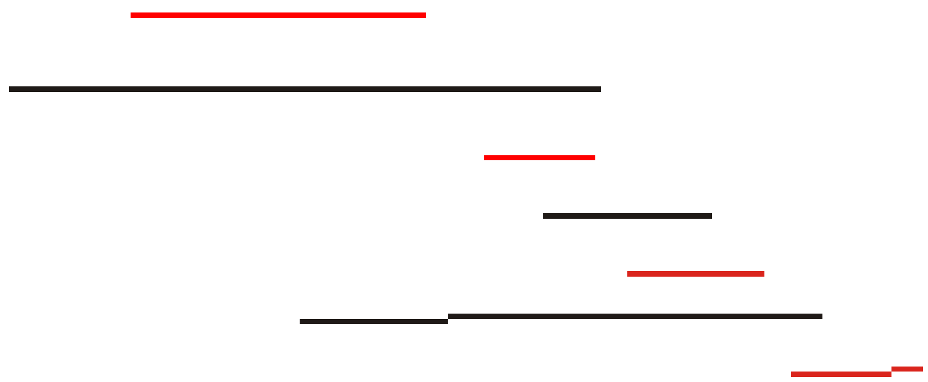
A megoldás egy N elemű halmaz legnagyobb, adott tulajdon-sággal rendelkező részhalmazának kiválasztása.

Probléma: egy N elemű halmaznak 2^N részhalmaza van.

Mohó stratégia

Ötlet:

Rendezzük sorba a filmeket befejezési idejük szerint növekvő sorrendbe!



Ha a leghamarabb befejeződőt választjuk, akkor lesz a legtöbb lehetőségünk a többi közül választani.

Mohó stratégia

Megoldás-1:

Filmek (események) száma: N. Kezdőidők: K_i. Végidők: V_i.
Eredeti (rendezés előtti) sorszám: S_i.

```
Kiválogatás(N,K,V,Db,X):
  Rendezés(N,K,V,S)
  Db:=1; X(Db):=S(1); j:=1
  Ciklus i=2-től N-ig
    Ha K(i)≥V(j) akkor Db:=Db+1; X(Db):=S(i)
                       j:=i
  Ciklus vége
Eljárás vége.
```

Mohó stratégia

Megoldás-2:

Filmek (események) száma: N. Kezdőidők: K_i. Végidők: V_i. $Kezd_j$: a j-ben végződő, legkésőbb kezdődő film kezdete, S_j a sorszáma. $1 \le K_i, V_i \le M$.

```
Kiválogatás(N,K,V,Db,X):
  Db:=0; idő:=0; Kezdetek(N,K,V,Kezd,S)
  Ciklus i=1-től M-ig
    Ha S(i)≠0 és idő≤Kezd(i)
       akkor Db:=Db+1; X(Db):=S(i); idő:=i
  Ciklus vége
Eljárás vége.
```

Mohó stratégia

Megoldás-2:

Kezd$_j$ előállítása:

```
Kezdetek(N,K,V,Kezd,S):
  Kezd:=(0,…,0)
  Ciklus i=1-től N-ig
    Ha K(i)>Kezd(V(i))
       akkor Kezd(V(i)):=K(i); S(V(i)):=i
  Ciklus vége
Eljárás vége.
```

Mohó stratégia

Feladat:

Egy vállalkozó 1 napos munkákat vállal. Ismerjük mindegyik munka határidejét. N napot dolgozik, N igényt kapott. Egy nap csak 1 munkát végezhet. Add meg, hogy maximum hány munkát vállalhat el!

Megoldás:

A megoldás egy N elemű halmaz legnagyobb, adott tulajdon-sággal rendelkező részhalmazának kiválasztása.

Probléma: egy N elemű halmaznak 2^N részhalmaza van.

Mohó stratégia

Ötlet:

Tegyünk minden munkát a legutolsó napra, amikor még elvégezhető!

Ezzel a lehető legkevesebb másik munka elvállalását akadályozzuk meg.

Megoldás$_1$:

Vegyük sorra a munkákat és mindegyiknek keressük meg a határideje előtt utolsó szabad napot!

Mohó stratégia

Megoldás-1:

```
Kiválogatás(N,H,Db,Nap):
  Db:=0; Nap():=(0,…,0)
  Ciklus i=1-től N-ig
    Ciklus amíg H(i)>0 és Nap(H(i))>0
      H(i):=H(i)-1
    Ciklus vége
    Ha H(i)>0 akkor Db:=Db+1; Nap(H(i)):=i
  Ciklus vége
Eljárás vége.
```

Futási idő: $O(N^2)$

Mohó stratégia

Megoldás-2:

Rendezzük sorba a munkákat H(i) szerint! Egy munka elvégezhető a határidejére, ha kevesebbet választottunk ki előtte, mint a határideje. Tegyük a munkát az első szabad napra!

```
Kiválogatás(N,H,Db,Nap):
  Rendezés(N,H,S)
  Db:=1; Nap(Db):=S(1)
  Ciklus i=2-től N-ig
    Ha Db<H(i) akkor Db:=Db+1; Nap(Db):=S(i)
  Ciklus vége
Eljárás vége.
```

Futási idő: O(N)

Mohó stratégia

Feladat:

Egy vállalkozó 1 napos munkákat vállal. Ismerjük mindegyik munka határidejét. Egy nap csak 1 munkát végezhet. Az egyes munkákért különböző bért kaphat. Add meg, hogy maximum mennyit kereshet!

Megoldás:

A megoldás egy N elemű halmaz legnagyobb értékű, adott tulajdonsággal rendelkező részhalmazának kiválasztása.

Probléma: egy N elemű halmaznak 2^N részhalmaza van.

Mohó stratégia

Ötlet:

Rendezzük sorba a munkákat az összeg szerint csökkenő sorrendbe! Tegyünk minden munkát a legutolsó napra, amikor még elvégezhető!

Ezzel a lehető legkevesebb másik munka elvállalását akadályozzuk meg és csak nála olcsóbbakat.

Megoldás:

Vegyük sorra a munkákat és mindegyiknek keressük meg a határideje előtt utolsó szabad napot!

Mohó stratégia

Megoldás:

```
Kiválogatás(N,H,Ár,Db,Nap):
  Rendezés(N,H,Ár,S)
  Db:=0; Nap():=(0,…,0)
  Ciklus i=1-től N-ig
    Ciklus amíg H(i)>0 és Nap(H(i))>0
      H(i):=H(i)-1
    Ciklus vége
    Ha H(i)>0 akkor Db:=Db+1; Nap(H(i)):=S(i)
  Ciklus vége
Eljárás vége.
```

Futási idő: $O(N^2)$

Mohó stratégia

Feladat:

Egy rendezvényt olyan teremben tartanak, amelyben M ülőhely van. A szervezők megrendeléseket fogatnak, minden megrendelés egy [a,b] számpárral adható meg, ami azt jelenti, hogy a megrendelő olyan s ülőhelyet szeretne kapni, amelyre teljesül, hogy a≤s≤b.

Számítsd ki, hogy a szervező a megrendelések alapján a legjobb esetben hány megrendelést tud kielégíteni és adj is meg egy olyan jegykiosztást, amely kielégíti a megrendeléseket!

Mohó stratégia

Megoldás:

Adott x ülőhelyre jelölje H(x) azokat az igényeket, amelyeknek megfelelő az x sorszámú ülőhely: $H(x)=\{i \mid a_i \leq x \leq b_i\}$!

Mohó választás: Az ülőhelyek szerint növekvően haladva az x ülőhelyre válasszuk azt a megfelelő igénylőt, akihez tartozó intervallum jobb végpontja a legkisebb!

H egyetlen prioritási sor lehet (b_i szerint), amelybe akkor kerül be egy index, ha intervallum kezdődik; és akkor kerül ki belőle, ha intervallum végződik!

A bemenet legyen a_i szerint rendezett!

Mohó stratégia

Megoldás:

```
Ülőhelyek(M,N,Beoszt):
  H:=üres; i:=1
  Ciklus x=1-től M-ig
    H bővítése azon i-kkel, amelyekre a(i)=x
    Ha H≠üres
      akkor PrSorból(H,i); Beoszt(x):=i
    H azon elemeinek törlése, amelyekre x=b(i)
  Ciklus vége
Eljárás vége.
```

Mohó stratégia

```
H bővítése azon i-kkel, amelyekre a(i)=x:
  Ciklus amíg i≤N és a(i)=x
    PrSorba(H,i); i:=i+1
  Ciklus vége
Eljárás vége.
H azon elemeinek törlése, amelyekre x=b(i):
  Ciklus amíg nem üres?(H) és első(H)=x
    PrSorból(H,j)
  Ciklus vége
Eljárás vége.
```

Mohó stratégia

Feladat:

Egy rendezvényre N vendég érkezik. Ismerjük mindegyiknek az érkezési és a távozási idejét. A résztvevőket fényképeken szeretnénk megörökíteni. Add meg, hogy minimum hányszor kell fényképet készíteni!

Megoldás:

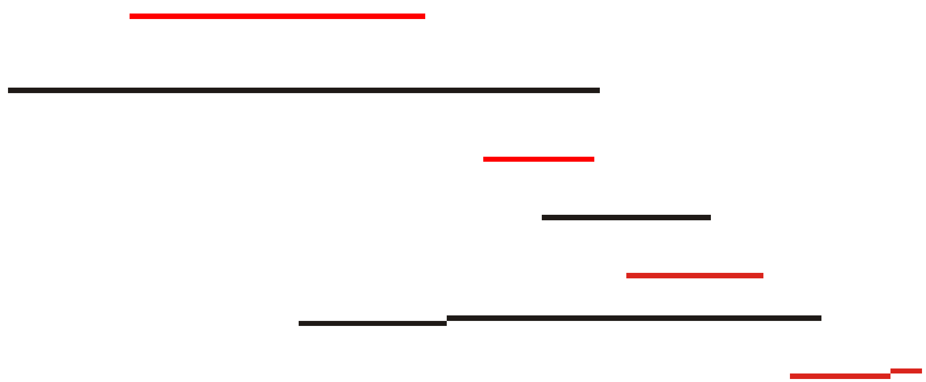
A megoldás a lehetséges időpontok halmaza legkisebb, adott tulajdonsággal rendelkező részhalmazának kiválasztása.

Probléma: egy N elemű halmaznak 2^N részhalmaza van.

Mohó stratégia

Megoldás:

Akkor fényképezzünk, amikor feltétlenül szükséges! Ez azt jelenti, hogy amikor elmenne az első ember, aki még nem volt rajta egy fényképen sem, akkor fényképeznünk kell.



Mohó stratégia

Megoldás:

Emberek (események) száma: N. Érkezési idők: E_i. Távozási idők: T_i. Eredeti (rendezés előtti) sorszám: S_i.

```
Kiválogatás(N,E,T,Db,X):
  Rendezés(N,E,T,S)
  Db:=1; X(Db):=S(1); j:=1
  Ciklus i=2-től N-ig
    Ha E(i)≥T(j) akkor Db:=Db+1; X(Db):=S(i)
                       j:=i
  Ciklus vége
Eljárás vége.
```

A megoldás szó szerint azonos a filmes feladat megoldásával!

Mohó stratégia

Feladat:

Egy rendezvényre N vendég érkezik. Ismerjük mindegyiknek az érkezési és a távozási idejét. A résztvevőket fényképeken szeretnénk megörökíteni. A fényképészt K perces időintervallumokra fizetjük. Add meg, hogy minimum hány intervallumra kell fizetni!

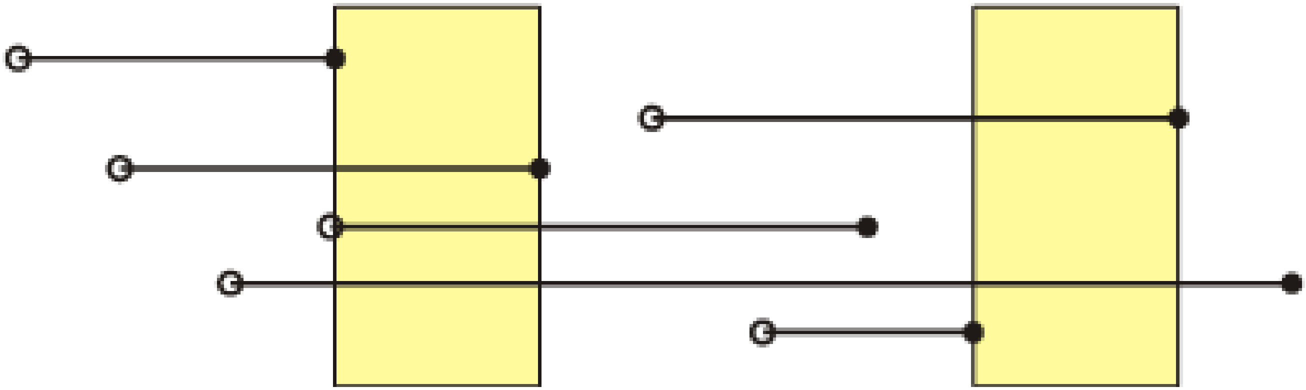
Megoldás:

A megoldás a lehetséges időpontok halmaza legkisebb, adott tulajdonsággal rendelkező részhalmazának kiválasztása.

Mohó stratégia

Megoldás:

Akkor fényképezzünk, amikor feltétlenül szükséges! Ez azt jelenti, hogy amikor elmenne az első ember, aki még nem volt rajta egy fényképen sem, akkor kezdődik egy fényképezési intervallum.



Mohó stratégia

Megoldás:

Emberek (események) száma: N. Érkezési idők: E_i. Távozási idők: T_i. Eredeti (rendezés előtti) sorszám: S_i.

```
Kiválogatás(N,E,T,K,Db,X):
  Rendezés(N,E,T,S)
  Db:=1; X(Db):=S(1); j:=1
  Ciklus i=2-től N-ig
    Ha E(i)≥T(j)+K akkor Db:=Db+1; X(Db):=S(i)
                         j:=i
  Ciklus vége
Eljárás vége.
```

Mohó stratégia

Feladat:

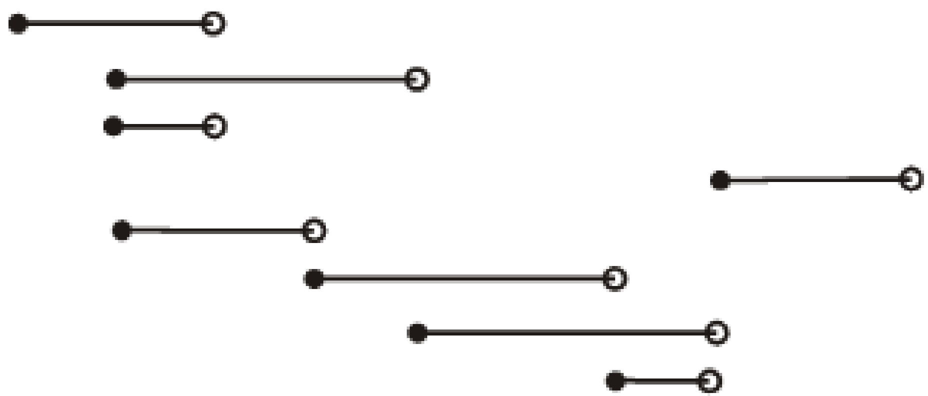
Egy rendezvényre N vendég érkezik. Ismerjük mindegyiknek az érkezési és a távozási idejét. A résztvevőket fényképeken szeretnénk megörökíteni, két feltétellel:

1. Minden képen pontosan két vendég legyen rajta.
2. Minden vendég legfeljebb egy képen szerepelhet.

Add meg, hogy maximum hányan lehetnek rajta a képeken!

Megoldás:

Rendezzük sorba távozási idő szerint az adatainkat. Ha vala-kinek még nincs párja a fényképezkedéshez, akkor válasszunk olyat párjának a korábban elmenők közül, aki ott van és a leghamarabb menne el!



Mohó stratégia

Emberek (események) száma: N. Érkezési idők: E_i. Távozási idők: T_i. Eredeti (rendezés előtti) sorszám: S_i.

```
Kiválogatás(N,E,T,Db,Pár):
  Rendezés(N,E,T,S); Db:=0
  Ciklus i=2-től N-ig
    j:=1
    Ciklus amíg j<i és nem(Pár(j)=0 és E(i)<T(j))
      j:=j+1
    Ciklus vége
    Ha j<i akkor Pár(j):=i; Pár(i):=j; Db:=Db+1
  Ciklus vége
Eljárás vége.
```

Mohó stratégia

A mohó stratégia elemei

1. Fogalmazzuk meg az optimalizációs feladatot úgy, hogy választások sorozatával építjük fel a megoldást!

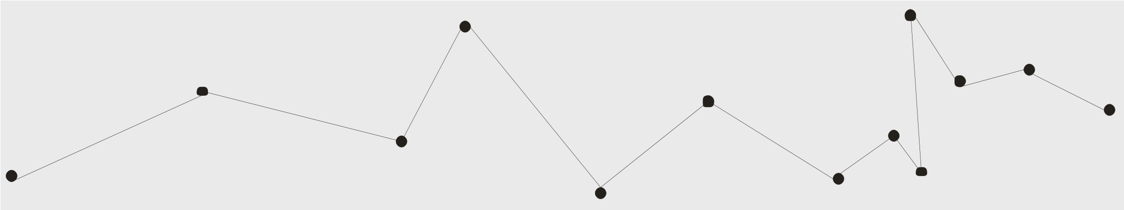
2. Mohó választási tulajdonság: Mutassuk meg, hogy mindig van olyan megoldása az eredeti feladatnak, amely a mohó választással kezdődik!

3. Optimális részprobléma tulajdonság: Bizonyítsuk be, hogy a mohó választással olyan redukált problémát kapunk, amelynek optimális megoldásához hozzávéve a mohó választást, az eredeti probléma megoldását kapjuk!

Mohó stratégia

Benzinkút probléma

Egy útvonalon N benzinkút található. Ismerjük az egyes benzinkutak távolságát, valamint azt, hogy tele tankkal az autónk hány kilométert tud megtenni (K)! Számold ki, hogy minimum hány helyen kell tankolnunk, s mondd is meg, hogy mely benzinkutaknál!



Mohó stratégia

A megoldás a benzinkutak halmazának egy $B_1, \ldots, B_k$ részhalmaza, ahol $Táv(B_{i+1})-Táv(B_i) \le K$ és $Táv(N)-Táv(B_k) \le K$.

Ami nem kérdés: a kezdőpontban (azaz az 1. benzinkútnál) tankolni kell, azaz $B_1=1$.

Ami szintén nem kérdéses: a célpontban már nem kell tankolni!

Mohó választás: Mindig a lehető legkésőbb tankoljunk, azaz B_2 legyen az a benzinkút, amelyre $Táv(B_2)-Táv(B_1) \le K$, de $Táv(B_2+1)-Táv(B_1) > K$.

Mohó stratégia

Bizonyítás: Ha korábban (valamely j sorszámú benzinkútnál) tankolnánk, akkor 2 lehetőség van:

Táv(B_3)-Táv(j)$\leq$K $\rightarrow$ ugyanolyan számú tankolással célba érhetünk.

Táv(B_3)-Táv(j)>K $\rightarrow$ másodszor is hamarabb kell tankolnunk, ekkor a megoldás a további tankolási helyektől függ,

…

Azaz ha nem mohó módon választunk, akkor a tankolások száma vagy nem változik, vagy nagyobb lesz!

Mohó stratégia

```
Benzinkút(N,Táv,db,B):
  db:=1; B(db):=1
  Ciklus i=2-től N-1-ig
    Ha Táv(i+1)-Táv(B(db))>K
       akkor db:=db+1; B(db):=i
  Ciklus vége
Eljárás vége.
```

Megjegyzés: Ha Táv(N)-Táv(B(db))>K, akkor nincs megoldás.

Mohó stratégia

Staféta

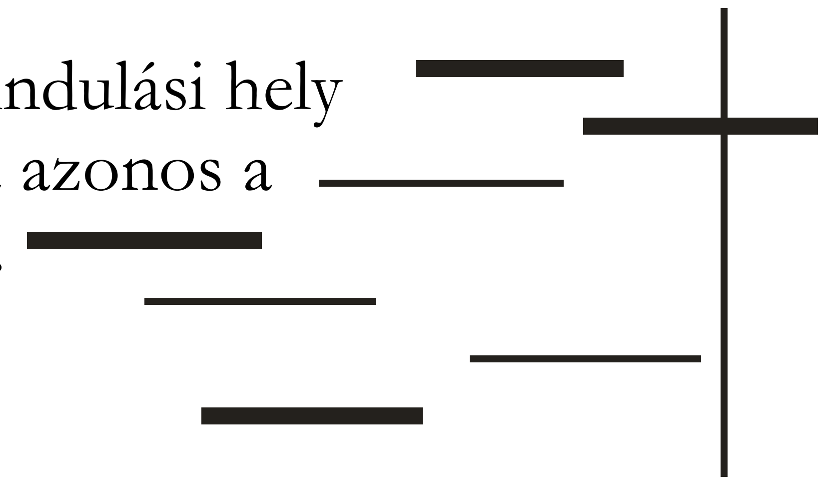
Az olimpiai lángot egy kiindulási városból a cél városba kell eljuttatni. A két város távolsága K kilométer. A szervezők meghirdették, hogy olyan futók jelentkezését várják, akik pontosan H kilométert futnak az olimpiai lánggal. Sok futó jelentkezett, mindegyik megadta, hogy hányadik kilométertől vállalja a futást. A szervezők ki akarják választani a jelentkezők közül a lehető legkevesebb futót, akik végigviszik a lángot.

Mohó stratégia

Ha egy futó az x kilométertől fut, akkor minden olyan futó át tudja venni tőle a lángot, aki olyan z kilométertől vállalja a futást, hogy $z \leq x+H$.

A kiindulási városból biztosan indulni kell egy futónak. A megoldás a futók egy olyan $F_1, \ldots, F_k$ részhalmaza, amikor minden futó a lehető legkésőbb adja át a lángot a következő futónak.

Ha sorba rendezzük a futókat az indulási hely szerint, akkor a feladat megoldása azonos a benzinkutas feladat megoldásával.



Mohó stratégia

```
Staféta(N,E,H,db,B):
  db:=1; B(db):=1
  Ciklus i=2-től N-1-ig
    Ha E(i+1)>E(B(db))+H
       akkor db:=db+1; B(db):=i
  Ciklus vége
Eljárás vége.
```

Megjegyzés: Ha E(N)>E(B(db))+K, akkor nincs megoldás.

Mohó stratégia

Staféta

Az olimpiai lángot egy kiindulási városból a cél városba kell eljuttatni. A két város távolsága K kilométer. Sok futó jelentkezett, mindegyikről tudjuk, hogy hányadik kilométertől hányadik kilométerig vállalja a futást. A szervezők ki akarják választani a jelentkezők közül a lehető legkevesebb futót, akik végigviszik a lángot.

Mohó stratégia

Ha egy futó az ***x*** kilométertől az ***y*** kilométerig vállalja a futást, akkor minden olyan futó át tudja venni tőle a lángot, aki olyan ***z*** kilométertől vállalja a futást, hogy $x \le z \le y$.

A kiindulási városból biztosan indulni kell egy futónak. A megoldás a futók egy olyan $F_1, \ldots, F_k$ részhalmaza, amikor minden futó annak adja át a lángot, aki a lehető legtovább tudja vinni.

Mohó stratégia

Az i-edik futó E(i) kilométertől V(i) kilométerig vállalja a láng vitelét.

Rendezzük sorba a futókat az indulási hely szerint!

Az utoljára kiválasztott futó érkezési helyéig válasszuk ki azt a futót, aki a legmesszebb vinné a lángot! Ha a következő futó már később indul, mint az aktuális futó befejezné a futást, akkor a leg-messzebb menőnek kell átadnia a lángot.

Mohó stratégia

```
Staféta(N,E,V,db,B):
  db:=1; B(db):=1; lm:=1
  Ciklus i=2-től N-1-ig
    Ha V(i)>V(lm) akkor lm:=i
    Ha E(i+1)>V(B(db))
       akkor db:=db+1; B(db):=lm
  Ciklus vége
Eljárás vége.
```

Megjegyzés: Ha E(N)>V(B(db)), akkor nincs megoldás.

Pakol – mohó stratégia

Feladat:

Egy raktárban egyetlen hosszú sorban ládák vannak. Minden láda kocka alakú, de méretük különböző lehet. A ládák egymásra rakásával akarnak helyet felszabadítani. A biztonsági előírás szerint több ládát is lehet egymásra rakni, de minden ládát csak nála nagyobbra lehet helyezni. Az i-edik helyen lévő ládát csak akkor lehet rárakni a j-edik helyen lévő torony tetejére, ha az i-edik és j-edik helyek között már nincs láda (j lehet akár kisebb, akár nagyobb, mint i). Minden ládát legfeljebb egyszer lehet mozgatni.

Pakol – mohó stratégia

Megoldás:

Haladjunk balról jobbra, amíg a láda méret növekszik. Ezek biztosan ráthetők arra, ameddig elértünk, de a tőle jobbra csökkenő sorrendben levők is (hacsak nincs két egyforma a két oldalon).

Példa:

1 3 **5** 4 2 6 **8** 7 **6** 5 3 **4**

Pakol – mohó stratégia

```
Pakol(m):
  m:=0; i:=1; a[n+1]:=0
  Ciklus
    b:=i
    Ciklus amíg i<n és a[i]<a[i+1]
      i:=i+1
    Ciklus vége
    bb:=i-1; j:=i+1; t:=a[i]
    Ciklus amíg b≤bb
     Ha a[j]<t és a[bb]<a[j] akkor t:=a[j]; j:=j+1
     különben t:=a[bb]; bb:=bb-1
    Ciklus vége
    …
```

Pakol – mohó stratégia

```
Pakol(m):
    … {maradtak jobbra}
    Ciklus amíg j≤n és t>a[j]
      t:=a[j]; j:=j+1
    Ciklus vége
    m:=m+1; i:=j
  amíg i≤n
  Ciklus vége
Eljárás vége.
```

Rendezvény – mohó stratégia

Feladat:

Egy rendezvényen N előadást szeretnének tartani. Minden előadó megadta, hogy az előadását mettől meddig tartaná. Add meg, hogy minimum hány termet kell biztosítani az előadásoknak, hogy mindegyiket megtarthassák!

Megoldás:

Rendezzük sorba az előadásokat kezdési idő szerint! Vegyük sorra az előadásokat és tegyük be az első terembe, ahova betehetők! Ha mindegyik terem foglalt, akkor új termet kell kezdenünk!

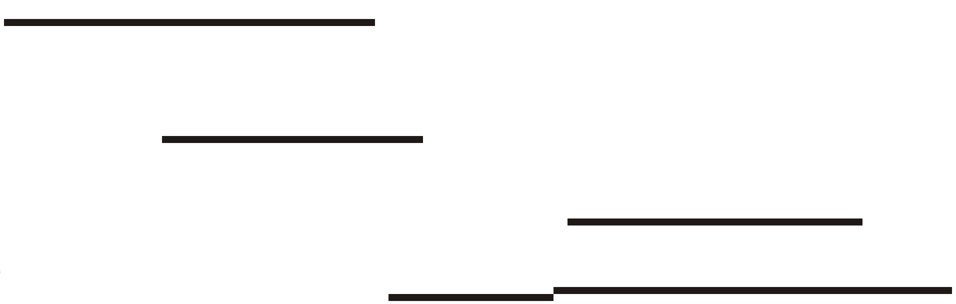
Rendezvény – mohó stratégia

```
Rendezvény(N,A,B):
  Db:=0
  Ciklus i=1-től N-ig
    j:=1
    Ciklus amíg j≤Db és A(i)<Határ(j)
      j:=j+1
    Ciklus vége
    Ha j≤Db akkor Határ(j):=B(i); Be(i):=j
         különben Db:=Db+1; Határ(Db):=B(i)
                  Be(i):=Db
  Ciklus vége
Eljárás vége.
```

Rendezvény – mohó stratégia

Megoldás:

Lehetne a legtöbb előadást berakni az első terembe, majd a maradékra ugyanezt alkalmazni (mint a filmes feladatnál)?



Nem! Ellenpélda:

A „filmes" megoldás szerint az 1. és a 3. kerülne az első terembe, a maradékból csak a 2. osztható be a második terembe, a 4.-nek új termet kell nyitni!

A jó megoldás: (1,4), (2,3) 2 teremben.

Lift - Mohó stratégia

Feladat:

Egy N emeletes házban szokatlan módon üzemeltetik a liftet. A lift az első szintről indul és mindig felmegy a legfelső szintre, majd visszatér az első szintre. Menet közben megáll minden olyan szinten, amelyik úticélja valamelyik liftben tartózkodó utasnak. Olyan szinten is megáll, ahonnan utazni szándékozik valaki az aktuális irányban, feltéve, hogy még befér a liftbe (figyelembe véve az adott szinten kiszállókat). A liftben egyszerre K ember lehet.

Legkevesebb hány menet (egyszer felmegy, majd lejön) szükséges ahhoz, hogy minden várakozó embert elszállítson a lift?

Lift – mohó stratégia

Legyen E(i,j) az i. szintről utazó j. ember célemelete (E,i,j)=0 az utolsó után)! Ezek alapján ki tudjuk számolni azt is, hogy az i. emeleten felfelé menő liftből hányan szállnának ki (CF(i)), illetve a lefelé menő liftből hányan szállnának ki (CL(i)).
Ezekből azt is kiszámolhatjuk, hogy az i. emeletről hányan mennének felfelé (F(i)), illetve lefelé (L(i)).
A mohó megoldás lényege: mindig a lehető legtöbb ember legyen a liftben!

$$\text{Menet} = \max_{i=1..n}\left(\max\begin{cases} (F(i)-1) \ \text{div} \ K+1 \\ (L(i)-1) \ \text{div} \ K+1 \end{cases} \right)$$

Lift – mohó stratégia

```
Lift:
  Menet:=0; F(0):=0
  Ciklus i=1-től N-ig
    F(i):=F(i-1)-CF(i); j:=1
    Ciklus amíg E(i,j)>0
      Ha E(i,j)>i akkor F(i):=F(i)+1
      j:=j+1
    Ciklus vége
    At:=(F(i)-1) div K+1
    Ha At>Menet akkor Menet:=At
  Ciklus vége
  …
```

Lift – mohó stratégia

```
  …
  L(N+1):=0
  Ciklus i=N-től 1-ig -1-esével
    L(i):=L(i+1)-CL(i); j:=1
    Ciklus amíg E(i,j)>0
      Ha E(i,j)<i akkor L(i):=L(i)+1
      j:=j+1
    Ciklus vége
    At:=(L(i)-1) div K+1
    Ha At>Menet akkor Menet:=At
  Ciklus vége
Eljárás vége.
```

Robotok - Mohó stratégia

Feladat:

Egy üzemben a gyártást automatizálták. A szerszámgépek egy nagy gépcsarnokban négyzetrács mentén vannak elhelyezve. A műszak végén robotok gyűjtik össze a szerszámgépek gyártotta alkatrészeket. A robotok négyzetrács alakú pályán mozognak a szerszámgépek fölötti térben. A négyzetrács bal felső sarkából, az (1,1) pontból indulnak, és a jobb alsó sarokba viszik el az alkatrészeket. A robotokat úgy tervezték, hogy csak jobbra és „lefelé” haladhatnak.

Minimálisan hány robotot kell elindítani az összes alkatrész begyűjtéséhez?

Robotok - Mohó stratégia

Keressük meg az aktuális oszlopban a legalsó 1-est!
Idáig egy robotnak biztos le kell jönnie, a korábbi oszlopokból a leglejjebb, de ennél a pontnál feljebb levő átjöhet ezt az 1-est is felszedni, s haladhatunk az oszlopban felfelé.
Ha újabb 1-est találunk, akkor meg kell nézni, hogy az előző oszlopokból jöhet-e át ide valamelyik robot, … és így tovább.
Legyen U(i)=1, ha az i-edik sorra kell robot!

```
0-1-1-1-0-0-1-0-1 0 0 0
0 1 0 0 0 1 0 0 0 0 0 0
0 0 0 1 0 0 0 0 1 0 0 0
1 0 0 0 0 1 0 0 0 0 0 0
0 0 0 1-0 0 0 0 0 0 0 0
0 1 0 0 1-0 0 0 0 0 0 0
1 0 0 0 0 1-1-1-1 0 0 0
1 1-1-1-1-1-0-0-0-1 0 0
1 0 0 0 0 0 0 0 0 1 0 0
1-0-0-0-0-0-0-0-0-1-0-0
```

Robotok - Mohó stratégia

```
Robotok:
  U():=(1,0,…,0)
  Ciklus j=1-től M-ig
    i:=N+1; U(0):=1
    Ciklus amíg i>0
      Ciklus
        i:=i-1
      amíg i>0 és T(i,j)≠1
      Ciklus vége
      …
```

Robotok - Mohó stratégia

```
      …
      Ha i>0 akkor ii:=i
                   Ciklus amíg U(ii)=0
                     ii:=ii-1
                   Ciklus vége
                   U(ii):=0; U(i):=1; i:=ii
    Ciklus vége
  Ciklus vége
  Megold:=0
  Ciklus i=1-től M-ig
    Megold:=Megold+U(i)
  Ciklus vége
Eljárás vége.
```

Taxi - Mohó stratégia

Feladat:
Egy taxi vállalkozó N megálló között szállít utasokat minibusszal. A korlátozások előírták neki, hogy egy menetben mindig az 1. megállótól kell indulnia és az i-edik megállótól az i+1-edik megállóba kell mennie. Ismeri az utasok igényeit, tehát minden utasról tudja, hogy melyik megállótól melyik megállóig akar utazni.
Legjobb esetben összesen hány utast tud egy menetben az utas igényének megfelelő helyre elszállítani?

Taxi - Mohó stratégia

Minden állomáson legfeljebb K utas lehet a taxiban.
Ehhez minden állomáson adjuk meg a K hely kiszállási idejét!
A taxiba beszállásról a kiszállási idő pillanatában döntünk.
Az i. állomáson kiszálló utas akkor szállhat be a taxiba, ha van olyan hely a taxiban, amiről az utolsó utas az ő beszállási ideje előtt szállt ki. Ha több ilyen is van, akkor azt kell választani, aki közülük a legkésőbb szállt ki!

Rendezzük sorba a bemenetet kiszállási idő szerint növekvő sorrendbe, azon belül pedig beszállási idő szerint növekvőbe!

Taxi - Mohó stratégia

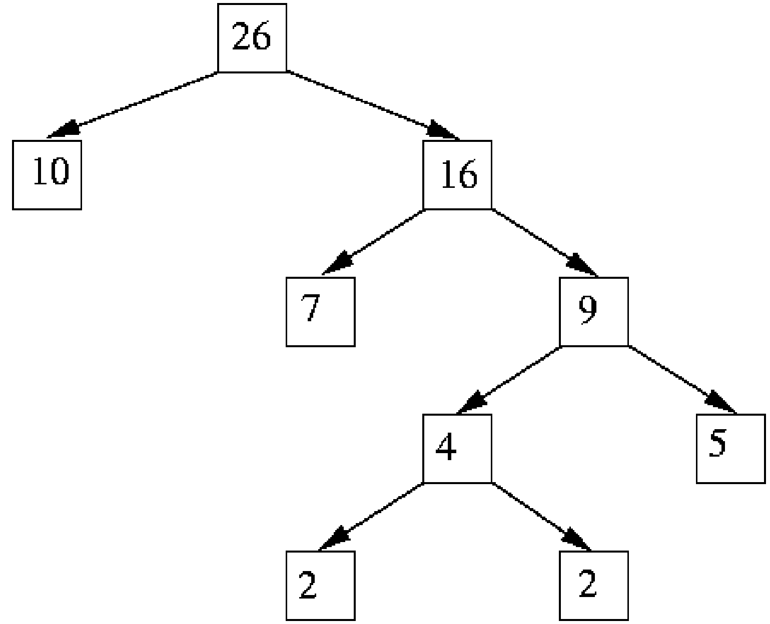
```
Taxi:
  Rendezések; Hely:=(0,...,0)
  Ciklus i=1-től N-ig
    Ért:=0; Max:=0
    Ciklus j=1-től K-ig
      Ha Hely(j)≤Be(i) akkor
         Ha Hely(j)>Ért akkor Ért:=Hely(j); Max:=j
    Ciklus vége
    Ha Max>0 akkor Db:=Db+1; Hely(Max):=Ki(i)
  Ciklus vége
Eljárás vége.
```

Darabolás - Mohó stratégia

Feladat:
Adott egy fémrúd, amelyet megadott számú és hosszúságú darabokra kell felvágni. Olyan vágógéppel kell a feladatot megoldani, amely egyszerre csak egy vágást tud végezni. A vágások tetszőleges sorrendben elvégezhetőek. Egy vágás költsége megegyezik annak a darabnak a hosszával, amit éppen (két darabra) vágunk. A célunk optimalizálni a műveletsor teljes költségét.

Darabolás - Mohó stratégia

Minden darabolás, így az optimális is leírható egy bináris fával. A fa levelei tartalmazzák a bemenetként kapott darabok hosszait, és minden belső pontja annak a darabnak a hosszát, amelyből vágással a két fiú-pontban lévő darab keletkezett, azaz a két fiának az összegét. Példánk esetén a fa a következőképpen néz ki.



Darabolás - Mohó stratégia

A darabolás összköltsége is kifejezhető a fával, nevezetesen, az összköltség éppen a fa belső (nem levél) pontjaiban található számok összege. Fordítva is igaz, minden ilyen fa egy darabolást ír le. A fa költségén a fa belső pontjaiban lévő számok összegét értjük. Tehát keressük az optimális megoldást, mint egy darabolási fát, tehát azt, amelynek a költsége minimális. A darabolási fa költsége kifejezhető a következőképpen. Legyenek $d_1, \ldots , d_N$ a vágandó darabok hosszai és legyen m_i a d_i. darabot tartalmazó levélpont mélysége (a fa gyökerétől vett távolsága) a fában. Ezekkel a jelölésekkel a fa költsége:

$$\sum_{i=1}^{N} m_i * d_i$$

Darabolás - Mohó stratégia

Az optimális fára teljesül:

- A két legkisebb értéket tartalmazó levélpont mélysége a legnagyobb, és testvérek.
- A két legkisebbet elhagyva optimális megoldást kapunk arra a bemenetre, amiben a két legkisebb helyett az összegük szerepel.

Építsük a darabolási fát úgy, hogy lépésenként a két legkisebb értéket tartalmazó pontot egy új pont két fiává tesszük, és az új pontba a két fiúban lévő érték összegét írjuk!

Darabolás - Mohó stratégia

```
Darabol:
  Költség:=0
  Ciklus i=1-től N-ig
    PrSorba(i); Fa(i).bal:=0; Fa(i).jobb:=0
  Ciklus vége
  Ciklus i=1-től N-1-ig
    PrSorból(x); PrSorból(y)
    z:=i+N; D(z):=D(x)+D(y); PrSorba(z)
    Fa(z).bal:=x; Fa(z).jobb:=y
    Költség:=Költség+D(z)
  Ciklus vége
Eljárás vége.
```

Konténer – Mohó stratégia

Feladat

Egy raktárban egyetlen sorban egymás mellett van N≤10000 kocka alakú konténer. Mindegyik konténer egy konténerhelyet foglal el a K méretétől (K≤1000) függetlenül. A raktárosnak helyet kell biztosítani újonnan érkező konténerek számára. Helyet csak úgy tud biztosítani, ha konténereket egymásra rak. Konténer csak nálánál nagyobb méretű konténerre rakható rá, de ennek betartásával akárhány konténer rakható egymásra. Az átpakolást olyan robottal végezhetjük, amely bármely konténert fel tud venni, de csak balról jobbra haladva tud szállítani.

Konténer – Mohó stratégia

Megoldás

Minden mérethez adjuk meg az olyan méretű konténerek indexei sorozatát, növekvő sorrendben!

Induljunk méret szerint visszafelé és minden előző helyen levő kisebb konténerből egyet tegyünk az aktuálisra!

Legyen Db(i) az i méretű konténerek száma, Index(i,j) pedig a j. i méretű konténer sorszáma!

Jelölje Akt(i) azt a j indexet, amelyik konténert az i méretűek közül még nem raktuk sehova!

Taxi - Mohó stratégia

```
Konténer:
  Előkészítés(m,Db,Index)
  Akt(i):=(1,...,1); Hely:=0;
  Ciklus amíg m>0
    i:=Index(m,Akt(m))
    Ciklus k=m-1-től 1-ig -1-esével
      Ha Akt(k)≤Db(k) és Akt(k)<i
         akkor Hely:=Hely+1; Akt(k):=Akt(k)+1
    Ciklus vége
    Akt(m):=Akt(m)+1
    Ciklus amíg m>0 és Akt(m)>Db(m)
      m:=m-1
    Ciklus vége
  Ciklus vége
Eljárás vége.
```

Mohó stratégia

A mohó stratégia elemei

1. Fogalmazzuk meg az optimalizációs feladatot úgy, hogy választások sorozatával építjük fel a megoldást!

2. Mohó választási tulajdonság: Mutassuk meg, hogy mindig van olyan megoldása az eredeti feladatnak, amely a mohó választással kezdődik!

3. Optimális részprobléma tulajdonság: Bizonyítsuk be, hogy a mohó választással olyan redukált problémát kapunk, amelynek optimális megoldásához hozzávéve a mohó választást, az eredeti probléma megoldását kapjuk!

Mohó stratégia

További példák

Egy kultúrháznak két nagy előadóterme van, A és B. Egy napon sok előadást szeretnének tartani a két teremben. Az igazgató begyűjtötte az igényeket, azt, hogy ki mettől-meddig akar előadást tartani. Egy teremben egyszerre csak egy előadás tartható. Ha egy előadás az T időpontban ér véget, akkor a következő előadás legkorábban a T+1 időpontban kezdődhet.

Készíts programot, amely kiszámítja a legtöbb előadás számát, amelyek megtarthatók a két teremben és megad egy beosztást a két teremre!

További példák

Egy folyó mentén N település helyezkedik el, melyek közül néhányon vízirendőrséget szeretnének alapítani. Egy rendőrcsónak a folyó folyásirányában A, folyásiránnyal szemben pedig B kilométert tud megtenni óránként.

Készíts programot, amely megadja azt a minimális számú települést, ahol vízirendőrséget kell létrehozni úgy, hogy a rendőrségekről bármely település K órán belül elérhető legyen!

Mohó stratégia

További példák

A Rendőrség szemtanúkat keres egy rendezvényen történt gyanús események kivizsgálásához. A rendezvény szervezői feljegyezték minden vendégről, hogy mikor érkezett és mikor távozott.

Készíts programot, amely megadja, hogy minimálisan hány vendéget kell kiválasztani, hogy minden gyanús esemény időpontjához legyen olyan kiválasztott vendég, aki jelen volt az esemény időpontjában!

Mohó stratégia

További példák

A DélKert szövetkezetben N termelő termel gyümölcsöt, amit a szövetkezet két hűtőházban gyűjt össze. Az i-edik termelő a leszedett gyümölcsöt az A hűtőházba ai, a B hűtőházba bi idő alatt tudja beszállítani. Az A hűtőház N1, a B pedig N2 termelőtől tud gyümölcsöt fogadni.

Készíts programot, amely kiszámítja, hogy az egyes termelőknek melyik hűtőházba kell szállítnia a leszedett gyümölcsöt, hogy az összes termelőtől a gyümölcs a lehető legkorábban hűtőházba kerüljön!

Mohó stratégia

További példák

Egy gazdának van néhány útja, amelyek mentén nyárfákat ültetett. Minden két szomszédos nyárfa közé pontosan egy szilvafát ültetett.

A gazda halála előtt a következőt mondta legidősebb fiának: kiválaszthatsz magadnak N darab nyárfát, és a tiéd lesz minden két szomszédos nyárfád közötti szilvafa is!

Készíts programot, amely az utak és a kiválasztható nyárfák számának ismeretében kiszámítja, hogy a fiúnak a legjobb esetben hány szilvafája lesz!

Mohó stratégia