Feladatmegoldási stratégiák
Oszd meg és uralkodj!
Visszalépéses keresés

Feladatmegoldási stratégiák

Oszd meg és uralkodj!

Több részfeladatra bontás, amelyek hasonlóan oldhatók meg, lépései:

- a triviális eset (amikor nincs rekurzív hívás)
- felosztás (megadjuk a részfeladatokat, amikre a feladat lebontható)
- uralkodás (rekurzívan megoldjuk az egyes részfeladatokat)
- összevonás (az egyes részfeladatok megoldásából előállítjuk az eredeti feladat megoldását)

Feladatmegoldási stratégiák
Oszd meg és uralkodj

Ezek alapján a következőképpen fogunk gondolkodni:

- Mi a leállás (triviális eset) feltétele? Hogyan oldható meg ilyenkor a feladat?
- Mi az általános feladat alakja? Mik a paraméterei? Ebből kapjuk meg a rekurzív eljárásunk specifikációját.
- Milyen paraméterértékekre kapjuk a konkrét feladatot? Ezekre fogjuk meghívni kezdetben az eljárást!

Feladatmegoldási stratégiák
Oszd meg és uralkodj

Ezek alapján a következőképpen fogunk gondolkodni:

- Hogyan vezethető vissza a feladat hasonló, de egyszerűbb részfeladatokra? Hány részfeladatra vezethető vissza?
- Melyek ilyenkor az általános feladat részfeladatainak a paraméterei? Ezekkel kell majd meghívni a rekurzív eljárást!
- Hogyan építhető fel a részfeladatok megoldásaiból az általános feladat megoldása?

Keresés rendezett sorozatban

Feladat:

Egy Y értéket keresünk egy rendezett X sorozatban.

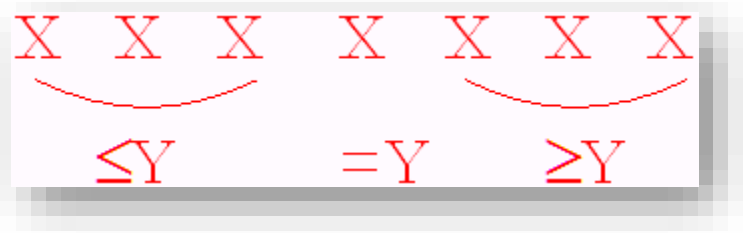
Ötlet és – tömb esetén – lehetőség:

Először a középső elemmel hasonlítsunk! Ha nem a keresett, akkor vagy előtte, vagy mögötte kell tovább keresni!

Logaritmikus keresés rendezett sorozatban

Itt akkor van megoldás, ha megtaláltuk a keresett érték valamelyikét.

```
Keresés:
   e:=1; u:=N
   Ciklus
      k:=(e+u) div 2
      Elágazás
         X[k]>Y esetén u:=k-1
         X[k]<Y esetén e:=k+1
      Elágazás vége
   amíg e≤u és X[k]≠Y
   Ciklus vége
   Van:=X[k]=Y
   …
Eljárás vége.
```



Feladatmegoldási stratégiák
Oszd meg és uralkodj

Gyorsrendezés (quicksort):

- felbontás: $\boxed{X_1, \ldots, X_{k-1}}\ X_k\ \boxed{X_{k+1}, \ldots, X_n}$ szétválogatás

  ahol $\forall i,j\ (1 \le i \le k;\ k \le j \le n)$: $X_i \le X_j$
- uralkodás: mindkét részt ugyanazzal a módszerrel felbontjuk két részre, rekurzívan
- összevonás: automatikusan történik a helyben szétválogatás miatt
- triviális eset: $n \le 1$

Feladatmegoldási stratégiák
Oszd meg és uralkodj

Gyorsrendezés (quicksort):

```
Quick(E,U):
   Szétválogatás(E,U,K)
   Ha E<K-1 akkor Quick(E,K-1)
   Ha k+1<U akkor Quick(K+1,U)
Eljárás vége.
```

Feladatmegoldási stratégiák
Oszd meg és uralkodj

Gyorsrendezés (quicksort):

```
Szétválogatás(E,U,K):
  segéd:=X(E)
  Ciklus amíg E<U
    Ciklus amíg E<U és segéd≤X(U)
      U:=U-1
    Ciklus vége
    ...
```

Véletlenített algoritmus:

v:=véletlen(E..U)

segéd:=X(v); X(v):=X(E)

Feladatmegoldási stratégiák
Oszd meg és uralkodj

Gyorsrendezés (quicksort):

```
   ...
   Ha E<U akkor X(E):=X(U); E:=E+1
                Ciklus amíg E<U és segéd≥X(E)
                  E:=E+1
                Ciklus vége
                Ha E<U akkor X(U):=X(E); U:=U-1
   Elágazás vége
  Ciklus vége
  X(E):=segéd; K:=E
Eljárás vége.
```

Feladatmegoldási stratégiák
Oszd meg és uralkodj

Összefésüléses rendezés (mergesort):

- felbontás: a sorozat két részsorozatra bontása (középen)

  $X_1, \dots, X_k$ | $X_{k+1}, \dots, X_n$

- uralkodás: a két részsorozat rendezése (rekurzívan)
- összevonás: a két rendezett részsorozat összefésülése
- triviális eset: $n \leq 1$

Feladatmegoldási stratégiák
Oszd meg és uralkodj

Összefésüléses rendezés (mergesort):

```
Rendez(E,U):
   Ha E<U akkor K:=(E+U)/2
                Rendez(E,K); Rendez(K+1,U)
                Összefésül(E,K,U)
Eljárás vége.
```

Feladatmegoldási stratégiák
Oszd meg és uralkodj

Összefésüléses rendezés (mergesort):

```
Összefésül(E,K,U):
  i:=1; j:=1; hova:=E-1
  Y():=X(E..K)⊕+∞; Z():=X(K+1..U)⊕+∞
  Ciklus amíg i<K-E+2 vagy j<U-K+1
    hova:=hova+1
    Ha Y(i)<Z(j) akkor  X(hova):=Y(i); i:=i+1
                különben X(hova):=Z(j); j:=j+1
  Ciklus vége
Eljárás vége.
```

Mivel a helyben összefésülés nagyon bonyolult lenne, ezért a két összefésülendőt kimásoljuk, majd onnan fésülünk össze.

Feladatmegoldási stratégiák
Oszd meg és uralkodj

i-edik legkisebb kiválasztása:

- felbontás: $\boxed{X_1, \ldots, X_{k-1}}\ X_k\ \boxed{X_{k+1}, \ldots, X_n}$ szétválogatás (ahol $\forall i,j\ (1 \le i \le k;\ k \le j \le n)$: $X_i \le X_j$)
- uralkodás: i<K esetén az első, i>K esetén a második részben keresünk tovább, rekurzívan
- összevonás: automatikusan történik a helyben szétválogatás miatt
- triviális eset: i=k

Feladatmegoldási stratégiák
Oszd meg és uralkodj

i-edik legkisebb kiválasztása:

```
Kiválasztás(E,U,i,Y):
   Szétválogatás(E,U,K)
   Ha i=K akkor Y:=X(K)
   különben ha i<K akkor Kiválasztás(E,K-1,i,Y)
   különben Kiválasztás(K+1,U,i-K,Y)
Eljárás vége.
```

Jó esetben N+N/2+N/4+…, azaz
N*(1+1/2+1/4+…)<2*N művelet!

Feladatmegoldási stratégiák
Oszd meg és uralkodj

Nem mindig két részre osztjuk a problémát:

Egy számkitalálós játékban, amiben egy 1 és N közötti szám kitalálása a cél, egyetlen típusú kérdést tehetünk fel: a kitalálandó szám az [A,B] intervallumon belül van-e? Erre háromféle választ kaphatunk:

- -1, ha a kitalálandó szám A-nál kisebb,
- 0, ha a kitalálandó szám az [A,B] intervallumban van
- +1, ha a kitalálandó szám B-él nagyobb

A lehető legkevesebb lépésben határozd meg, hogy mi a kitalálandó szám!

Feladatmegoldási stratégiák
Oszd meg és uralkodj

```
Kitalál(e,u):
   Ha e=u akkor Kitalál:=E
   különben a:=(2*e+u)/3; b:=(e+2*u)/3
            j:=kérdés(a,b)
            Elágazás
             j=-1 esetén Kitalál:=Kitalál(e,a-1)
             j=0 esetén Kitalál:=Kitalál(a,b)
             j=1 esetén Kitalál:=Kitalál(b+1,u)
            Elágazás vége
Eljárás vége.
```

Itt tehát az [e,u] intervallumot három részre osztottuk.

Feladatmegoldási stratégiák
Visszalépéses keresés

Visszalépéses keresés (backtrack)

A visszalépéses keresés lényege a feladat megoldásának előállítása rendszeres próbálgatással.

Adott N sorozat, amelyek rendre M(1), M(2), ... M(N) elemszámúak. Ki kell választani mindegyikből egy-egy elemet úgy, hogy az egyes sorozatokból való választások másokat befolyásolnak. Másképp fogalmazva: egy adott tulajdonsággal rendelkező szám N-est kell megadni úgy, hogy ne kelljen az összes lehetőséget végignézni!

Feladatmegoldási stratégiák
Visszalépéses keresés

N vezér elhelyezése egy NxN-es sakktáblán

Helyezzünk el egy NxN-es sakktáblán N vezért úgy, hogy ne üssék egymást!

Egy lehetséges megoldás N=5-re és N=4-re:

		V		
				V
	V			
			V	
V				

	V		
			V
V			
		V	

Feladatmegoldási stratégiák
Visszalépéses keresés

Munkásfelvétel: N állás – N jelentkező

Egy vállalkozás N különböző állásra keres munkásokat. Pontosan N jelentkező érkezett, ahol minden jelentkező megmondta, hogy mely munkákhoz ért. A vállalkozás vezetője azt szeretné, ha az összes jelentkezőt fel tudná venni és minden munkát el tudna végeztetni.

	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

A visszalépéses keresés megoldási elve

E feladatok közös jellemzője, hogy eredményük egy sorozat.

E sorozat minden egyes tagját valamilyen sorozatból kell kikeresni (vezért egy oszlop valamely helyére, egy munkásnak a vállalt munkák közül valamelyiket), de az egyes keresések összefüggenek egymással (vezért nem lehet oda tenni, ahol egy korábban letett vezér ütné; egy munkát nem lehet két munkásnak adni).

Feladatmegoldási stratégiák
Visszalépéses keresés

A visszalépéses keresés megoldási elve

- A visszalépéses keresés olyan esetekben használható, amikor a keresési tér fastruktúraként képzelhető el, amiben a gyökérből kiindulva egy csúcsot keresünk.
- Az algoritmus lényege, hogy a kezdőpontból kiindulva megtesz egy utat a feladatot részproblémákra bontva, és ha valahol az derül ki, hogy már nem juthat el a célig, akkor visszalép egy korábbi döntési ponthoz, és ott más utat – más részproblémát választ.

Feladatmegoldási stratégiák
Visszalépéses keresés

Először megpróbálunk az első sorozatból kiválasztani egy elemet, ezután a következőből, ...

Ha nincs jó választás, akkor visszalépünk az előző sorozathoz, s megpróbálunk abból egy másik elemet választani.

Visszalépésnél törölni kell a választást abból a sorozatból, amelyikből visszalépünk. Az eljárás akkor ér véget, ha minden sorozatból sikerült választani, vagy pedig a visszalépések sokasága után már az első sorozatból sem lehet újabb elemet választani (ekkor a feladatnak nincs megoldása).

Feladatmegoldási stratégiák
Visszalépéses keresés

Visszalépéses keresés algoritmus:

```
Keresés(N,Van,Y):
  i:=1; Y:=(0,...,0)
  Ciklus amíg i≥1 és i≤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.
```

A megoldás legfelső szintjén keressünk az i. sorozatból megfelelő elemet! Ha ez sikerült, akkor lépjünk tovább az i+1. sorozatra, különben lépjünk vissza az i-1.-re, s keressünk abban újabb elemet!

Feladatmegoldási stratégiák
Visszalépéses keresés

Visszalépéses keresés algoritmus:

```
Jóesetkeresés(i,Van,j):
  j:=Y(i)+1
  Ciklus amíg j≤M(i) és
                        (rossz(i,j) vagy tilos(j))
    j:=j+1
  Ciklus vége
  Van:=(j≤M(i))
Eljárás vége.
```

Megjegyzés: az i-edik lépésben a j-edik döntési út nem választható, ha az előzőek miatt rossz, vagy ha önmagában rossz.

Feladatmegoldási stratégiák
Visszalépéses keresés

Visszalépéses keresés algoritmus:

```
rossz(i,j):                                  {1. változat}
  k:=1
  Ciklus amíg k<i és szabad(i,j,k,Y(k))
    k:=k+1
  Ciklus vége
  rossz:=(k<i)
Eljárás vége.
```

Megjegyzés: Rossz egy választás, ha valamelyik korábbi választás miatt nem szabad – eldöntés.

Feladatmegoldási stratégiák
Visszalépéses keresés

Visszalépéses keresés algoritmus:

```
rossz(i,j):                                  {2. változat}
  s:=F0
  Ciklus k=1-től i-1-ig
    s:=f(s,k,Y(k))
  Ciklus vége
  rossz:=nem szabad(s,i,j)
Eljárás vége.
```

Megjegyzés: Rossz egy választás, ha a korábbiak összessége miatt nem szabad – sorozatszámítás, megszámolás, …

Feladatmegoldási stratégiák
Visszalépéses keresés

Feladat

Helyezzünk el egy NxN-es sakktáblán N vezért úgy, hogy ne üssék egymást!

A vezérek a sorukban, az oszlopukban és az átlójukban álló bábukat üthetik. Tehát úgy kell elhelyezni a vezéreket, hogy minden sorban és minden oszlopban is pontosan 1 vezér legyen, és minden átlóban legfeljebb 1 vezér legyen!

Feladatmegoldási stratégiák
Visszalépéses keresés

N vezér a sakktáblán:

```
Keresés(N,Van,Y):
  i:=1; Y:=(0,...,0)
  Ciklus amíg i≥1 és i≤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 vezér a sakktáblán:

```
Jóesetkeresés(i,Van,j):
  j:=Y(i)+1
  Ciklus amíg j≤N és rossz(i,j)
    j:=j+1
  Ciklus vége
  Van:=(j≤N)
Eljárás vége.
```

Feladatmegoldási stratégiák
Visszalépéses keresés

N vezér a sakktáblán:

```
rossz(i,j):
  k:=1
  Ciklus amíg k<i és Y(k)≠j és i-k≠abs(j-Y(k))
    k:=k+1
  Ciklus vége
  rossz:=(k<i)
Eljárás vége.
```

Feladatmegoldási stratégiák
Oszd meg és uralkodj!
Visszalépéses keresés