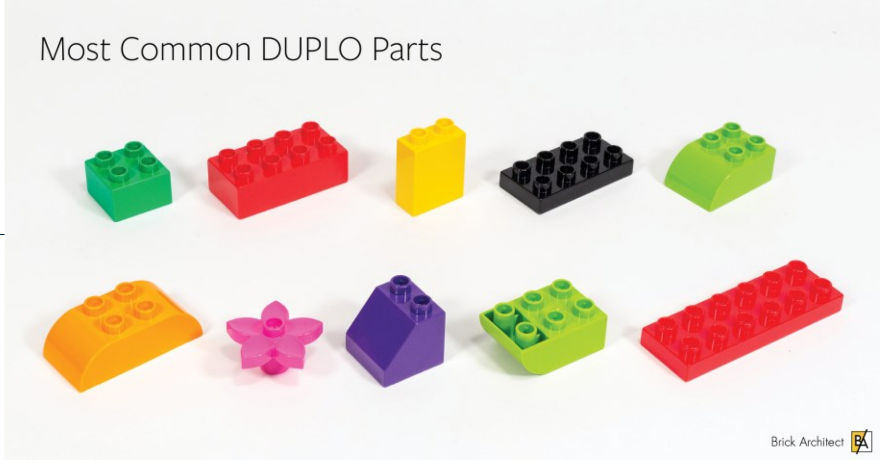
ELTE | IK

PROGRAMOZÁS

Tanulságos feladatok

Horváth Győző

Programozási minták



1. Összegzés
2. Megszámolás
3. Maximumkiválasztás
    a. Minimumkiválasztás
4. Feltételes maximumkeresés
    a. Feltételes minimumkeresés
5. Keresés
6. Eldöntés
    a. Mind eldöntés
7. Kiválasztás
8. Másolás
9. Kiválogatás

Tanulságos feladatok

Függvény két visszatérési értékkel

Feladat:

Egy térképet egy négyzetrácsban tárolunk. Minden rácspontban egy egész szám mondja meg az ott lévő pont tengerszint feletti magasságát. Határozd meg a térképen a legészaknyugatibb pontot, ahol tó található!

40	74	42	20	10	51	61	23	6	89	75	46	40	28	27	79	91	33	99	27	51	92
54	44	53	0	70	14	72	72	59	87	66	82	26	66	78	65	55	91	59	77	67	73
5	16	0	0	0	0	24	70	49	90	89	43	92	9	62	3	89	56	21	47	50	25
38	69	35	0	0	0	0	19	88	19	99	1	80	##	27	54	46	77	55	32	46	31
59	43	56	0	0	26	5	86	56	92	49	88	18	67	81	45	85	26	79	74	41	36
57	13	80	69	38	35	32	25	89	43	48	20	62	81	49	94	6	88	47	2	57	91
12	22	21	41	12	52	84	95	40	13	48	36	55	22	50	48	64	52	##	73	98	49
86	57	70	14	80	98	36	45	49	32	28	53	58	99	36	40	93	90	82	74	98	14
57	12	45	36	75	36	4	58	64	54	87	3	0	0	0	57	75	26	44	30	35	43
80	10	65	1	1	19	12	69	38	24	6	62	0	0	22	27	85	73	31	2	87	56
9	89	18	54	37	94	78	80	16	21	93	96	85	0	8	81	30	87	1	75	80	40
##	83	70	63	2	29	47	29	68	24	4	50	96	92	95	90	29	97	90	86	11	99
10	4	60	64	31	55	72	84	69	12	43	28	35	74	18	11	54	64	73	48	60	76

1. változat

Feladat:

Egy egész számokat tartalmazó mátrixban melyik az a sor és oszlop, ahol sorfolytonosan először 0 szerepel?

Másképpen:

Egy egész számokat tartalmazó mátrixban melyik az a sor, amelyben van 0, és ebben hol fordul elő először?

Azaz:

Keresésben eldöntés és kiválasztás

40	74	42	20	10	51	61	23	6	89	75	46	40	28	27	79	91	33	99	27	51	92
54	44	53	0	70	14	72	72	59	87	66	82	26	66	78	65	55	91	59	77	67	73
5	16	0	0	0	0	24	70	49	90	89	43	92	9	62	3	89	56	21	47	50	25
38	69	35	0	0	0	0	19	88	19	99	1	80	##	27	54	46	77	55	32	46	31
59	43	56	0	0	26	5	86	56	92	49	88	18	67	81	45	85	26	79	74	41	36
57	13	80	69	38	35	32	25	89	43	48	20	62	81	49	94	6	88	47	2	57	91
12	22	21	41	12	52	84	95	40	13	48	36	55	22	50	48	64	52	##	73	98	49
86	57	70	14	80	98	36	45	49	32	28	53	58	99	36	40	93	90	82	74	98	14
57	12	45	36	75	36	4	58	64	54	87	3	0	0	0	57	75	26	44	30	35	43
80	10	65	1	1	19	12	69	38	24	6	62	0	0	22	27	85	73	31	2	87	56
9	89	18	54	37	94	78	80	16	21	93	96	85	0	8	81	30	87	1	75	80	40
##	83	70	63	2	29	47	29	68	24	4	50	96	92	95	90	29	97	90	86	11	99
10	4	60	64	31	55	72	84	69	12	43	28	35	74	18	11	54	64	73	48	60	76

40	74	42	20	10	51	61	23	6	89	75	46	40	28	27	79	91	33	99	27	51	92
54	44	53	0	70	14	72	72	59	87	66	82	26	66	78	65	55	91	59	77	67	73
5	16	0	0	0	0	24	70	49	90	89	43	92	9	62	3	89	56	21	47	50	25
38	69	35	0	0	0	0	19	88	19	99	1	80	##	27	54	46	77	55	32	46	31
59	43	56	0	0	26	5	86	56	92	49	88	18	67	81	45	85	26	79	74	41	36
57	13	80	69	38	35	32	25	89	43	48	20	62	81	49	94	6	88	47	2	57	91
12	22	21	41	12	52	84	95	40	13	48	36	55	22	50	48	64	52	##	73	98	49
86	57	70	14	80	98	36	45	49	32	28	53	58	99	36	40	93	90	82	74	98	14
57	12	45	36	75	36	4	58	64	54	87	3	0	0	0	57	75	26	44	30	35	43
80	10	65	1	1	19	12	69	38	24	6	62	0	0	22	27	85	73	31	2	87	56
9	89	18	54	37	94	78	80	16	21	93	96	85	0	8	81	30	87	1	75	80	40
##	83	70	63	2	29	47	29	68	24	4	50	96	92	95	90	29	97	90	86	11	99
10	4	60	64	31	55	72	84	69	12	43	28	35	74	18	11	54	64	73	48	60	76

1. változat

Feladat:

Egy egész számokat tartalmazó mátrixban melyik az a sor, amelyben van 0, és ebben hol fordul elő először?

Specifikáció:

```
Be: n∈N, m∈N, mátrix∈Z[1..n,1..m]
Ki: van∈L, sind∈N, oind∈N
Fv: vannulla:N->L, vannulla(i)=VAN(j=1..m, mátrix[i,j]=0)
Ef: -
Uf: (van,sind)=KERES(i=1..n, vannulla(i)) és
    van -> oind=KIVÁLASZT(j>=1, mátrix[sind,j]=0)
```

40	74	42	20	10	51	61	23	6	89	75	46	40	28	27	79	91	33	99	27	51	92
54	44	53	0	70	14	72	72	59	87	66	82	26	66	78	65	55	91	59	77	67	73
5	16	0	0	0	0	24	70	49	90	89	43	92	9	62	3	89	56	21	47	50	25
38	69	35	0	0	0	0	19	88	19	99	1	80	##	27	54	46	77	55	32	46	31
59	43	56	0	0	26	5	86	56	92	49	88	18	67	81	45	85	26	79	74	41	36
57	13	80	69	38	35	32	25	89	43	48	20	62	81	49	94	6	88	47	2	57	91
12	22	21	41	12	52	84	95	40	13	48	36	55	22	50	48	64	52	##	73	98	49
86	57	70	14	80	98	36	45	49	32	28	53	58	99	36	40	93	90	82	74	98	14
57	12	45	36	75	36	4	58	64	54	87	3	0	0	0	57	75	26	44	30	35	43
80	10	65	1	1	19	12	69	38	24	6	62	0	0	22	27	85	73	31	2	87	56
9	89	18	54	37	94	78	80	16	21	93	96	85	0	8	81	30	87	1	75	80	40
##	83	70	63	2	29	47	29	68	24	4	50	96	92	95	90	29	97	90	86	11	99
10	4	60	64	31	55	72	84	69	12	43	28	35	74	18	11	54	64	73	48	60	76

1. változat

Specifikáció:

```
Be: n∈N, m∈N, mátrix∈Z[1..n,1..m]
Ki: van∈L, sind∈N, oind∈N
Fv: vannulla:N->L, vannulla(i)=VAN(j=1..m, mátrix[i,j]=0)
Ef: -
Uf: (van,sind)=KERES(i=1..n, vannulla(i)) és
    van -> oind=KIVÁLASZT(j>=1, mátrix[sind,j]=0)
```

Keresés

```
ind    ~ sind
e..u   ~ 1..n
T(i)   ~ vannulla(i)
```

Eldöntés (vannulla)

```
i      ~ j
e..u   ~ 1..m
T(i)   ~ mátrix[i,j]=0
```

Kiválasztás

```
ind    ~ oind
i      ~ j
e      ~ 1
T(i)   ~ mátrix[sind,j]=0
```

1. változat

40	74	42	20	10	51	61	23	6	89	75	46	40	28	27	79	91	33	99	27	51	92
54	44	53	0	70	14	72	72	59	87	66	82	26	66	78	65	55	91	59	77	67	73
5	16	0	0	0	0	24	70	49	90	89	43	92	9	62	3	89	56	21	47	50	25
38	69	35	0	0	0	0	19	88	19	99	1	80	##	27	54	46	77	55	32	46	31
59	43	56	0	0	26	5	86	56	92	49	88	18	67	81	45	85	26	79	74	41	36
57	13	80	69	38	35	32	25	89	43	48	20	62	81	49	94	6	88	47	2	57	91
12	22	21	41	12	52	84	95	40	13	48	36	55	22	50	48	64	52	##	73	98	49
86	57	70	14	80	98	36	45	49	32	28	53	58	99	36	40	93	90	82	74	98	14
57	12	45	36	75	36	4	58	64	54	87	3	0	0	0	57	75	26	44	30	35	43
80	10	65	1	1	19	12	69	38	24	6	62	0	0	22	27	85	73	31	2	87	56
9	89	18	54	37	94	78	80	16	21	93	96	85	0	8	81	30	87	1	75	80	40
##	83	70	63	2	29	47	29	68	24	4	50	96	92	95	90	29	97	90	86	11	99
10	4	60	64	31	55	72	84	69	12	43	28	35	74	18	11	54	64	73	48	60	76

```
Uf: (van,sind)=KERES(i=1..n, vannulla(i)) és
    van -> oind=KIVÁLASZT(j>=1, mátrix[sind,j]=0)
```

Keresés

```
ind    ~ sind
e..u   ~ 1..n
T(i)   ~ vannulla(i)
```

Eldöntés (vannulla)

```
i      ~ j
e..u   ~ 1..m
T(i)   ~ mátrix[i,j]=0
```

Kiválasztás

```
ind    ~ oind
i      ~ j
e      ~ 1
T(i)   ~ mátrix[sind,j]=0
```

```
sind:=1
sind≤n és nem vannulla(sind)
    sind:=sind+1
van:=sind≤n
T                van                F
oind:=1                            -
nem mátrix[sind,j]=0
    oind:=oind+1
```

```
vannulla(i:Egész): Logikai

j:=1
j≤m és nem mátrix[i,j]=0
    j:=j+1
van:=j≤m
vannulla:=van
```

```
Változó
 j:Egész,
 van:Logikai
```

2. változat

Feladat:

Egy egész számokat tartalmazó mátrixban melyik az a sor és oszlop, ahol sorfolytonosan először 0 szerepel?

Másképpen:

Egy egész számokat tartalmazó mátrixban melyik az a sor, amelyben van 0, és ebben hol fordul elő először?

Azaz:

Keresésben keresés és keresés

40	74	42	20	10	51	61	23	6	89	75	46	40	28	27	79	91	33	99	27	51	92
54	44	53	0	70	14	72	72	59	87	66	82	26	66	78	65	55	91	59	77	67	73
5	16	0	0	0	0	24	70	49	90	89	43	92	9	62	3	89	56	21	47	50	25
38	69	35	0	0	0	0	19	88	19	99	1	80	##	27	54	46	77	55	32	46	31
59	43	56	0	0	26	5	86	56	92	49	88	18	67	81	45	85	26	79	74	41	36
57	13	80	69	38	35	32	25	89	43	48	20	62	81	49	94	6	88	47	2	57	91
12	22	21	41	12	52	84	95	40	13	48	36	55	22	50	48	64	52	##	73	98	49
86	57	70	14	80	98	36	45	49	32	28	53	58	99	36	40	93	90	82	74	98	14
57	12	45	36	75	36	4	58	64	54	87	3	0	0	0	57	75	26	44	30	35	43
80	10	65	1	1	19	12	69	38	24	6	62	0	0	22	27	85	73	31	2	87	56
9	89	18	54	37	94	78	80	16	21	93	96	85	0	8	81	30	87	1	75	80	40
##	83	70	63	2	29	47	29	68	24	4	50	96	92	95	90	29	97	90	86	11	99
10	4	60	64	31	55	72	84	69	12	43	28	35	74	18	11	54	64	73	48	60	76

2. változat

40	74	42	20	10	51	61	23	6	89	75	46	40	28	27	79	91	33	99	27	51	92
54	44	53	0	70	14	72	72	59	87	66	82	26	66	78	65	55	91	59	77	67	73
5	16	0	0	0	0	24	70	49	90	89	43	92	9	62	3	89	56	21	47	50	25
38	69	35	0	0	0	0	19	88	19	99	1	80	##	27	54	46	77	55	32	46	31
59	43	56	0	0	26	5	86	56	92	49	88	18	67	81	45	85	26	79	74	41	36
57	13	80	69	38	35	32	25	89	43	48	20	62	81	49	94	6	88	47	2	57	91
12	22	21	41	12	52	84	95	40	13	48	36	55	22	50	48	64	52	##	73	98	49
86	57	70	14	80	98	36	45	49	32	28	53	58	99	36	40	93	90	82	74	98	14
57	12	45	36	75	36	4	58	64	54	87	3	0	0	0	57	75	26	44	30	35	43
80	10	65	1	1	19	12	69	38	24	6	62	0	0	22	27	85	73	31	2	87	56
9	89	18	54	37	94	78	80	16	21	93	96	85	0	8	81	30	87	1	75	80	40
##	83	70	63	2	29	47	29	68	24	4	50	96	92	95	90	29	97	90	86	11	99
10	4	60	64	31	55	72	84	69	12	43	28	35	74	18	11	54	64	73	48	60	76

Feladat:

Egy egész számokat tartalmazó mátrixban melyik az a sor, amelyben van 0, és ebben hol fordul elő először?

```
keresnulla(i)=(van,ind):((van,ind)=KERES(j=1..m, mátrix[i,j]>0))
```

Specifikáció:

```
Be: n∈N, m∈N, mátrix∈Z[1..n,1..m]
Ki: van∈L, sind∈N, oind∈N
Fv: keresnulla:N->L x N,
    keresnulla(i)=KERES(j=1..m, mátrix[i,j]=0)
Ef: -
Uf: (van,sind)=KERES(i=1..n, keresnulla(i).van) és
    van -> oind=keresnulla(sind).ind
```

40	74	42	20	10	51	61	23	6	89	75	46	40	28	27	79	91	33	99	27	51	92
54	44	53	0	70	14	72	72	59	87	66	82	26	66	78	65	55	91	59	77	67	73
5	16	0	0	0	0	24	70	49	90	89	43	92	9	62	3	89	56	21	47	50	25
38	69	35	0	0	0	0	19	88	19	99	1	80	##	27	54	46	77	55	32	46	31
59	43	56	0	0	26	5	86	56	92	49	88	18	67	81	45	85	26	79	74	41	36
57	13	80	69	38	35	32	25	89	43	48	20	62	81	49	94	6	88	47	2	57	91
12	22	21	41	12	52	84	95	40	13	48	36	55	22	50	48	64	52	##	73	98	49
86	57	70	14	80	98	36	45	49	32	28	53	58	99	36	40	93	90	82	74	98	14
57	12	45	36	75	36	4	58	64	54	87	3	0	0	0	57	75	26	44	30	35	43
80	10	65	1	1	19	12	69	38	24	6	62	0	0	22	27	85	73	31	2	87	56
9	89	18	54	37	94	78	80	16	21	93	96	85	0	8	81	30	87	1	75	80	40
##	83	70	63	2	29	47	29	68	24	4	50	96	92	95	90	29	97	90	86	11	99
10	4	60	64	31	55	72	84	69	12	43	28	35	74	18	11	54	64	73	48	60	76

2. változat

Specifikáció:

```
Be: n∈N, m∈N, mátrix∈Z[1..n,1..m]
Ki: van∈L, sind∈N, oind∈N
Fv: keresnulla:N->L x N,
    keresnulla(i)=KERES(j=1..m, mátrix[i,j]=0)
Ef: -
Uf: (van,sind)=KERES(i=1..n, keresnulla(i).van) és
    van -> oind=keresnulla(sind).ind
```

Keresés

```
ind  ~ sind
e..u ~ 1..n
T(i) ~ keresnulla(i).van
```

Keresés (keresnulla)

```
i     ~ j
e..u  ~ 1..m
T(i)  ~ mátrix[i,j]=0
```

40	74	42	20	10	51	61	23	6	89	75	46	40	28	27	79	91	33	99	27	51	92
54	44	53	0	70	14	72	72	59	87	66	82	26	66	78	65	55	91	59	77	67	73
5	16	0	0	0	0	24	70	49	90	89	43	92	9	62	3	89	56	21	47	50	25
38	69	35	0	0	0	0	19	88	19	99	1	80	##	27	54	46	77	55	32	46	31
59	43	56	0	0	26	5	86	56	92	49	88	18	67	81	45	85	26	79	74	41	36
57	13	80	69	38	35	32	25	89	43	48	20	62	81	49	94	6	88	47	2	57	91
12	22	21	41	12	52	84	95	40	13	48	36	55	22	50	48	64	52	##	73	98	49
86	57	70	14	80	98	36	45	49	32	28	53	58	99	36	40	93	90	82	74	98	14
57	12	45	36	75	36	4	58	64	54	87	3	0	0	0	57	75	26	44	30	35	43
80	10	65	1	1	19	12	69	38	24	6	62	0	0	22	27	85	73	31	2	87	56
9	89	18	54	37	94	78	80	16	21	93	96	85	0	8	81	30	87	1	75	80	40
##	83	70	63	2	29	47	29	68	24	4	50	96	92	95	90	29	97	90	86	11	99
10	4	60	64	31	55	72	84	69	12	43	28	35	74	18	11	54	64	73	48	60	76

2. változat

Uf: (van,sind)=KERES(i=1..n, keresnulla(i).van) és
van -> oind=keresnulla(sind).ind

Keresés

ind ~ sind

e..u ~ 1..n

T(i) ~ keresnulla(i).van

Keresés (keresnulla)

i ~ j

e..u ~ 1..m

T(i) ~ mátrix[i,j]=0

sind:=1

sind≤n és nem keresnulla(sind).van

sind:=sind+1

van:=sind≤n

van

T

F

oind:=keresnulla(sind).ind

-

keresnulla(i:Egész): (Logikai,Egész)

ind:=1

ind≤m és nem mátrix[i,ind]=0

ind:=ind+1

van:=ind≤m

keresnulla:=(van,ind)

Változó

ind:Egé

van:Log

Maximumkiválasztás vs feltételes maximumkeresés

Feladat:

Egy gyártó bekéri egy üzlettől, hogy egy időszakon belül, amikor nyitva volt, hány darab fogyott egy adott termékből. Mennyi volt a legtöbb eladott darabszám?

5	3	0	0	2	1	7	8	4	3

Elemzés

5	3	0	0	2	1	7	8	4	3

Feladat:

Egy gyártó bekéri egy üzlettől, hogy egy időszakon belül, amikor nyitva volt, hány darab fogyott egy adott termékből. Mennyi volt a legtöbb eladott darabszám?

Maximumkiválasztás!

Specifikáció:

```
Be: n∈N, db∈N[1..n]
Ki: max∈N
Ef: -
Uf: (,max)=MAX(i=1..n, db[i])
```

Biztos???

Ha a sorozat lehet üres is, akkor nem biztos, hogy van maximális elem

→ feltételes maximumkeresés

1. változat

Feladat:

Egy gyártó bekéri egy üzlettől, hogy egy időszakon belül, amikor nyitva volt, hány darab fogyott egy adott termékből. Mennyi volt a legtöbb eladott darabszám?

Maximumkiválasztás!

Specifikáció:

```
Be: n∈N, db∈N[1..n]
Ki: van∈L, max∈N
Ef: -
Uf: van = n>0 és
    van -> (,max)=MAX(i=1..n, db[i])
```

2. változat

Feladat:

Egy gyártó bekéri egy üzlettől, hogy egy időszakon belül, amikor nyitva volt, hány darab fogyott egy adott termékből. Mennyi volt a legtöbb eladott darabszám?

Feltételes maximumkeresés

Specifikáció:

```
Be: n∈N, db∈N[1..n]
Ki: van∈L, max∈N
Ef: -
Uf: (van,,max)=FELTMAX(i=1..n, db[i], igaz)
```

Vezérlőelv

Feladat:

Egy polgárőrség nyilvántartásában m őr van, és tudjuk, hogy n nap mindegyikén melyik őr volt szolgálatban. Melyik őr volt a legtöbbször szolgálatban?

1.változat: nap-vezérelt megoldás

Feladat:

Egy polgárőrség nyilvántartásában m őr van, és tudjuk, hogy n nap mindegyikén melyik őr volt szolgálatban. Melyik őr volt a legtöbbször szolgálatban?

Lépések:

1. Minden naphoz meghatározzuk, hogy az adott napi őr, hányszor volt még szolgálatban
2. Ezek közül azt a napot választjuk, ahol ez a szám a legnagyobb
3. Maximumkiválasztásban megszámolás

1	3
2	5
3	4
4	5
5	4
6	6
7	5
n=8	3

1.változat: nap-vezérelt megoldás

Feladat:

Egy polgárőrség nyilvántartásában m őr van, és tudjuk, hogy n nap mindegyikén melyik őr volt szolgálatban. Melyik őr volt a legtöbbször szolgálatban?

Specifikáció:

```
Be: m∈N, n∈N, őr∈N[1..n]
Sa: ind∈N
Ki: terhelt∈N
Fv: hány:N->N,
    hány(nap)=DARAB(i=nap+1..n, őr[i]=őr[nap])
Ef: n>0
Uf: (ind,)=MAX(nap=1..n, hány(nap)) és
    terhelt=őr[ind] és m=m
```

1	3
2	5
3	4
4	5
5	4
6	6
7	5
n=8	3

2. változat: őr-vezérelt megoldás

Feladat:

Egy polgárőrség nyilvántartásában m őr van, és tudjuk, hogy n nap mindegyikén melyik őr volt szolgálatban. Melyik őr volt a legtöbbször szolgálatban?

nap	melyik őr		őrök=	1	2	3	4	5	6	7
1	5							x		
2	3					x				
3	4						x			
4	5							x		
5	4						x			
6	6								x	
7	3					x				
8	5							x		
			db=	0	0	2	2	3	1	0

2. változat: őr-vezérelt megoldás

Feladat:

Egy polgárőrség nyilvántartásában m őr van, és tudjuk, hogy n nap mindegyikén melyik őr volt szolgálatban. Melyik őr volt a legtöbbször szolgálatban?

Lépések:

1. Minden őrhöz meghatározzuk, hogy hányszor volt szolgálatban
2. Ezek közül azt az őrt választjuk, ahol ez a szám a legnagyobb
3. Maximumkiválasztásban megszámolás

őrök=	1	2	3	4	5	6	7
					x		
			x				
				x			
					x		
				x			
						x	
			x				
					x		
db=	0	0	2	2	3	1	0

őrök=	1	2	3	4	5	6	7
					x		
			x				
				x			
					x		
				x			
						x	
			x				
					x		
db=	0	0	2	2	3	1	0

2. változat: őr-vezérelt megoldás

Feladat:

Egy polgárőrség nyilvántartásában m őr van, és tudjuk, hogy n nap mindegyikén melyik őr volt szolgálatban. Melyik őr volt a legtöbbször szolgálatban?

Specifikáció:

```
Be: m∈N, n∈N, őr∈N[1..n]
Ki: terhelt∈N
Fv: hány:N->N,
    hány(melyikőr)=DARAB(i=1..n, őr[i]=melyikőr)
Ef: -
Uf: (terhelt,)=MAX(melyikőr=1..m, hány(melyikőr))
```

Korábban a másolási feladatról

- A másolás (és részben a kiválogatás) mindig összevonható az utána következő feladattal.

Másolással összeépítés

```
Be: n∈N, x∈H[1..n]
Ki: y∈H[1..n]
Ef: -
Uf: ∀i∈[1..n]:(y[i]=f(x[i]))
```
Rövidítve:
```
Uf: y=MÁSOL(i=e..u, f(x[i]))
```

A **másolás** programozási tétellel összeépítés minden programozási tételre működik. (sorozat→sorozat)

Csupán annyi a teendő, hogy a bemenetben szereplő sorozat-értékek helyett az i-edik feldolgozandó elemként a másolásban szereplő f-transzformáltat kell írni. Például:

Összegzéssel összeépítés:

```
SZUMMA(i=1..n,x[i]) → SZUMMA(i=1..n,f(x[i]))
```
vagy

Maximumkiválasztással összeépítés:

```
MAX(i=1..n,x[i]) → MAX(i=1..n,f(x[i]))
```

Itt tehát a „másik" tétel bemeneti sorozatára vonatkozik az f-transzformálás.

Adattranszformáció

Feladat:

Egy polgárőrség nyilvántartásában n őr van, és tudjuk minden őrről, hogy m napból mely naptól mely napig volt szolgálatban. Mikor volt a legvédettebb a helység?

Adattranszformáció

Feladat:

Egy polgárőrség nyilvántartásában n őr van, és tudjuk minden őrről, hogy m napból mely naptól mely napig volt szolgálatban. Mikor volt a legvédettebb a helység?

őr	tól	ig	nap=	1	2	3	4	5	6	7	8	9	10
1	1	3		x	x	x							
2	2	2			x								
3	5	8						x	x	x	x		
4	7	8								x	x		
n=5	8	9									x	x	
			db=	1	2	1	0	1	1	2	3	1	0

1. változat: nap-vezérelt megoldás

Feladat: Mikor volt a legvédettebb a helység?

Lépések:

Adattranszformáció másolással

1. Átalakítás: meghatározzuk minden naphoz, hogy akkor hány őr volt szolgálatban.
2. Melyik nap a legnagyobb a darabszám?
3. Másolásban megszámolás és maximumkiválasztás

őr	tól	ig	nap=	1	2	3	4	5	6	7	8	9	10
1	1	3		x	x	x							
2	2	2			x								
3	5	8						x	x	x	x		
4	7	8								x	x		
n=5	8	9									x	x	
			db=	1	2	1	0	1	1	2	3	1	0

1. változat: minden nap

őr	tól	ig
1	1	3
2	2	2
3	5	8
4	7	8
n=5	8	9

Feladat: Mikor volt a legvédettebb a helység?

Specifikáció:

```
Be: m∈N, n∈N, őrök∈Intervallum[1..n],
    Intervallum=(tól:N x ig:N)
Sa: db∈N[1..m]
Ki: maxind∈N
Fv: hányőr:N->N,
    hányőr(nap)=DARAB(i=1..n, őrök[i].tól<=nap<=őrök[i].ig)
Ef: m>0 és ∀i∈[1..n]:(őrök[i].tól>=1 és őrök[i].ig<=m)
Uf: ∀nap∈[1..m]:(db[nap]=hányőr(nap)) és
    (maxind,)=MAX(nap=1..m, db[nap])
```

nap=	1	2	3	4	5	6	7	8	9	10
	x	x	x							
		x								
					x	x	x	x		
							x	x		
								x	x	
db=	1	2	1	0	1	1	2	3	1	0

Nap-vezérelt megoldás

2. változat: másolás

Feladat: Mikor volt a legvédettebb a helység?

őr	tól	ig
1	1	3
2	2	2
3	5	8
4	7	8
n=5	8	9

Specifikáció:

```
Be: m∈N, n∈N, őrök∈Intervallum[1..n],
    Intervallum=(tól:N x ig:N)
Sa: db∈N[1..m]
Ki: maxind∈N
Fv: hányőr:N->N,
    hányőr(nap)=DARAB(i=1..n, őrök[i].tól<=nap<=őrök[i].ig)
Ef: m>0 és ∀i∈[1..n]:(őrök[i].tól>=1 és őrök[i].ig<=m)
Uf: db=MÁSOL(nap=1..m, hányőr(nap)) és
    (maxind,)=MAX(nap=1..m, db[nap])
```

nap=	1	2	3	4	5	6	7	8	9	10
	x	x	x							
		x								
					x	x	x	x		
							x	x		
								x	x	
db=	1	2	1	0	1	1	2	3	1	0

Érdemes megnézni, hogy a másolás összevonható-e a következő lépéssel!

3. változat: másolás összevonása

őr	tól	ig
1	1	3
2	2	2
3	5	8
4	7	8
n=5	8	9

Feladat: Mikor volt a legvédettebb a helység?

Specifikáció:

nap=	1	2	3	4	5	6	7	8	9	10
	x	x	x							
		x								
					x	x	x	x		
							x	x		
								x	x	
db=	1	2	1	0	1	1	2	3	1	0

```
Be: m∈N, n∈N, őrök∈Intervallum[1..n],
    Intervallum=(tól:N x ig:N)

Ki: maxind∈N
Fv: hányőr:N->N,
    hányőr(nap)=DARAB(i=1..n, őrök[i].tól<=nap<=őrök[i].ig)
Ef: m>0 és ∀i∈[1..n]:(őrök[i].tól>=1 és őrök[i].ig<=m)
Uf: (maxind,)=MAX(nap=1..m, hányőr(nap))
```

Adattranszformáció – mátrix, vezérlés

Feladat:

Egy busztársaság nyilvántartásában n buszjárat van, és mindegyikről tudjuk, hogy m város közül melyek között járnak buszok. Adj meg egy olyan várost, amelybe nem tudunk ezzel a busztársasággal utazni!

Adattranszformáció – mátrix, vezérlés

Feladat:

Egy busztársaság nyilvántartásában n buszjárat van, és mindegyikről tudjuk, hogy m város közül melyek között járnak buszok. Adj meg egy olyan várost, amelybe nem tudunk ezzel a busztársasággal utazni!

járatok	honnan	hova
1	3	6
2	5	7
3	2	3
4	2	5
5	1	6

városok=	1	2	3	4	5	6	7
1						x	
2			x		x		
3		x				x	
4							
5		x					x
6	x		x				
7					x		

Adattranszformáció – mátrix, vezérlés

Feladat:

Adj meg egy olyan várost, amelybe nem tudunk ezzel a busztársasággal utazni!

Lépések:

1. Átalakítás: meghatározzuk minden várospárhoz, hogy van-e a két város között járat.
2. Melyik oszlop üres?
3. Másolásban eldöntés és keresésben mind eldöntés

járatok	honnan	hova
1	3	6
2	5	7
3	2	3
4	2	5
5	1	6

városok=	1	2	3	4	5	6	7
1						x	
2			x		x		
3		x				x	
4							
5		x					x
6	x		x				
7					x		

1. változat: adattranszformáció

járatok	honnan	hova
1	3	6
2	5	7
3	2	3
4	2	5
5	1	6

Adj meg egy olyan várost, amelybe nem tudunk ezzel a busztársasággal utazni!

városok=	1	2	3	4	5	6	7
1						x	
2			x		x		
3		x				x	
4							
5		x					x
6	x		x				
7					x		

Specifikáció:

```
Be: m∈N, n∈N, járatok∈Járat[1..n],
    Járat=(honnan:N x hova:N)
Sa: mátrix∈L[1..m,1..m]
Ki: ind∈N
Fv: vanjárat:N x N->L,
    vanjárat(i,j)=VAN(k=1..n,
      (járatok[k].honnan=i és járatok[k].hova=j) vagy
      (járatok[k].hova=i és járatok[k].honnan=j))
Fv: üres:N->L,
    üres(hova)=MIND(honnan=1..m, mátrix[honnan,hova]=hamis)
Ef: ∀i∈[1..n]:
       (1<=járatok[i].honnan<=m és 1<=járatok[i].hova<=m)
Uf: ∀i∈[1..m]:(∀j∈[1..m]:(mátrix[i,j]=vanjárat(i,j))) és
    (,ind)=KERES(hova=1..m, üres(hova))
```

Érdemes megnézni, hogy a másolás összevonható-e a következő lépéssel!

2. változat: város-vezérelt megoldás

járatok	honnan	hova
1	3	6
2	5	7
3	2	3
4	2	5
5	1	6

Adj meg egy olyan várost, amelybe nem tudunk ezzel a busztársasággal utazni!

városok=	1	2	3	4	5	6	7
1						x	
2			x		x		
3		x				x	
4							
5		x					x
6	x		x				
7					x		

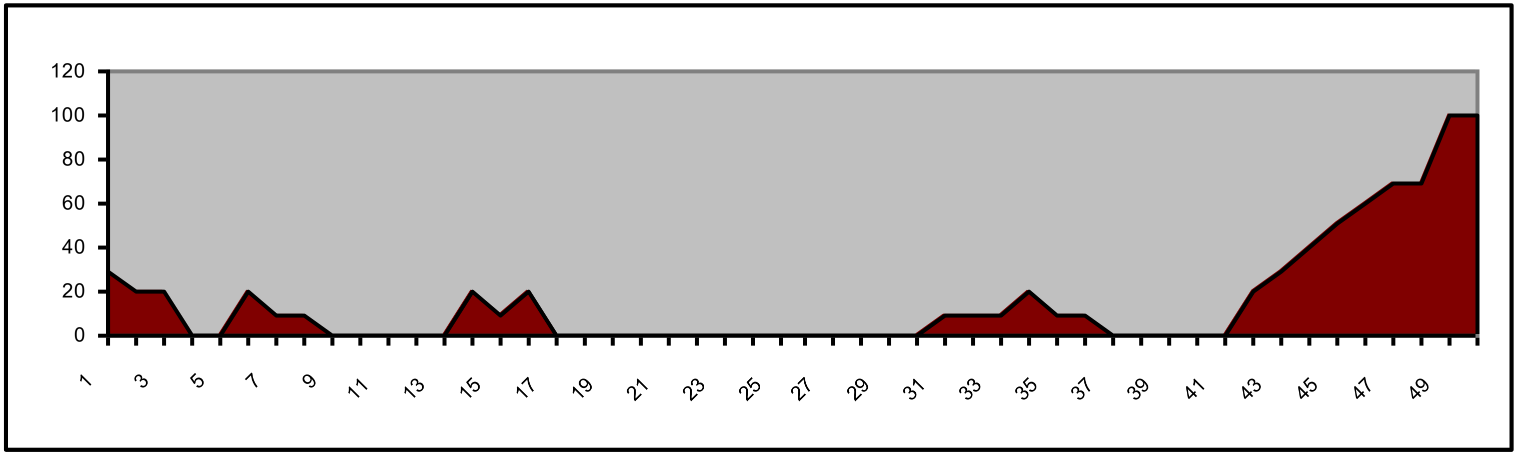
Specifikáció:

```
Be: m∈N, n∈N, járatok∈Járat[1..n],
    Járat=(honnan:N x hova:N)
Ki: ind∈N
Fv: vanjárat:N->L,
    vanjárat(város)=VAN(k=1..n,
           járatok[k].honnan=város vagy járatok[k].hova=város)
Ef: ∀i∈[1..n]:
            (1<=járatok[i].honnan<=m és 1<=járatok[i].hova<=m)
Uf: (,ind)=KERES(város=1..m, nem vanjárat(város))
```

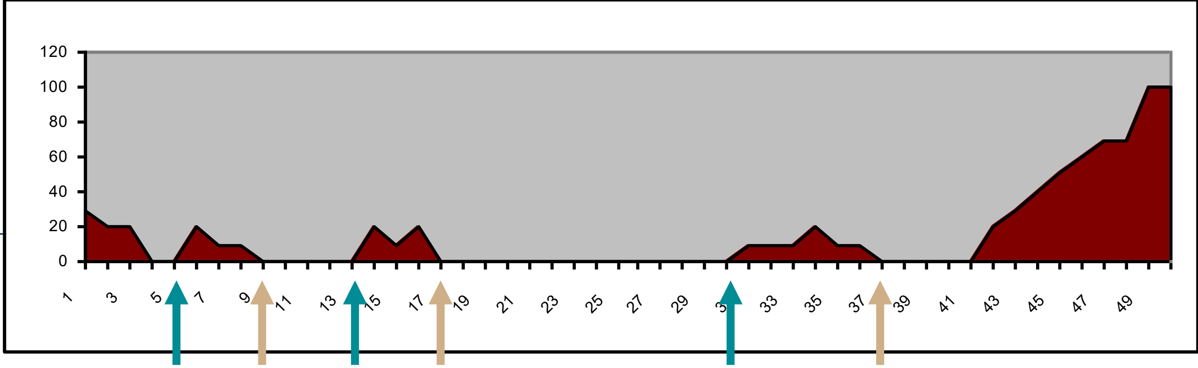
Intervallumos példák

Feladat:

Egy repülőgéppel Európából Amerikába repültünk. Az út során bizonyos kilométerenként mértük a felszín tengerszint feletti magasságát (≥0). 0 magasságot ott mértünk, ahol tenger van, >0-t pedig ott, ahol szárazföld. Adjuk meg a legszélesebb szigetet!



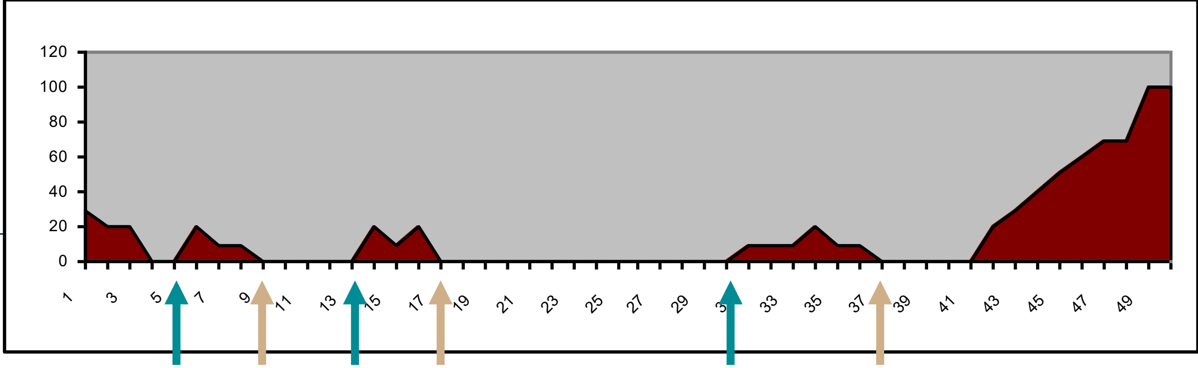
Intervallum-határok



Specifikáció:

```
Fv: szigetkezdet:N->L,
    szigetkezdet(i)=
      {hamis, ha i=1;
       mag[i-1]=0 és mag[i]>0 egyébként}
Fv: szigetvég:N->L,
    szigetvég(i)=
      {hamis, ha i=n;
       mag[i+1]=0 és mag[i]>0 egyébként}
```

1a. változat:

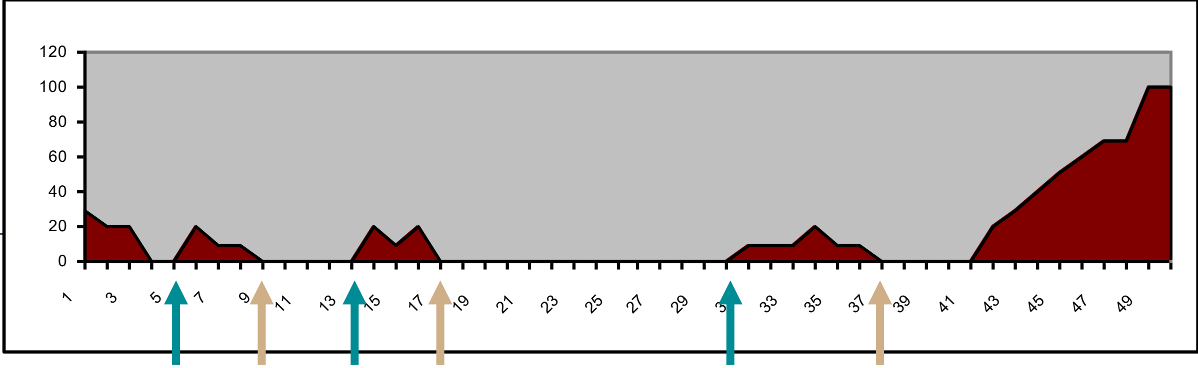


Specifikáció:

```
Be: n∈N, mag∈N[1..n]
Sa: db∈N, kezdetek∈N[1..db],
    végek∈N[1..db],
    távolságok∈N[1..db],
    maxind∈N
Ki: van∈L, k∈N, v∈N
Fv: ...
Fv: keresvége:N->L x N,
    keresvége(i)=KERES(j=i..n, szigetvég(j))
Ef: n>=3 és mag[1]>0 és mag[n]>0
Uf: (db,kezdetek)=KIVÁLOGAT(i=1..n,
                   szigetkezdet(i) és keresvége(i).van, i) és
    végek=MÁSOL(i=1..db, keresvége(kezdetek[i]).ind) és
    távolságok=MÁSOL(i=1..db, végek[i] - kezdetek[i] + 1) és
    (van,maxind,)=FELTMAX(i=1..db, távolságok[i], igaz) és
    van -> k=kezdetek[maxind] és v=végek[maxind]
```

1. Válogassuk ki azokat a szigetkezdeteket, amelyeknek van vége is!
2. Másolással határozzuk meg hozzájuk a szigetvégeket! (összevonható)
3. Másolással határozzuk meg a távolságokat! (összevonható)
4. Határozzuk meg a legnagyobb távolságot, ha van!

1b. változat:

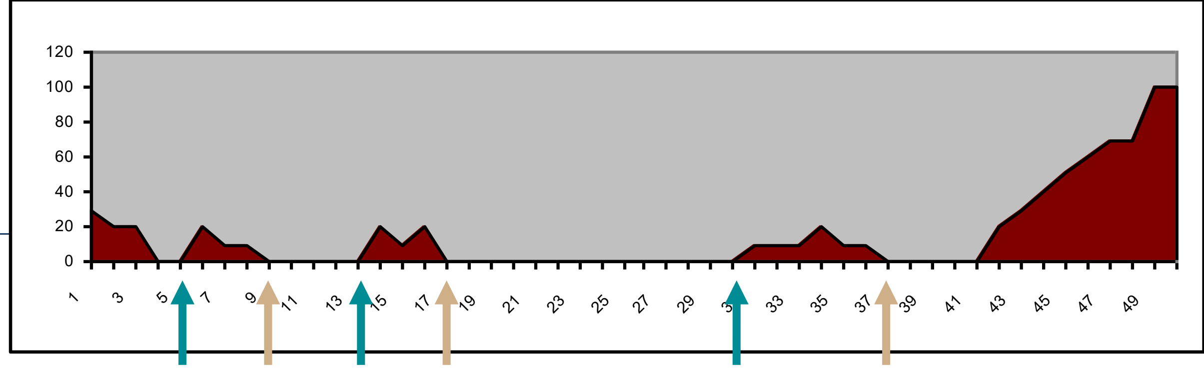


Specifikáció:

```
Be: n∈N, mag∈N[1..n]
Sa: db∈N, kezdetek∈N[1..db], maxind∈N
Ki: van∈L, k∈N, v∈N
Fv: szigetkezdet:N->L,
    szigetkezdet(i)={hamis, ha i=1;
                    mag[i-1]=0 és mag[i]>0 egyébként}
Fv: szigetvég:N->L,
    szigetvég(i)={hamis, ha i=n;
                 mag[i+1]=0 és mag[i]>0 egyébként}
Fv: keresvége:N->L x N,
    keresvége(i)=KERES(j=i..n, szigetvég(j))
Fv: táv:N->N, táv(i)=keresvége(i).ind - i
Ef: n>=3 és mag[1]>0 és mag[n]>0
Uf: (db,kezdetek)=KIVÁLOGAT(i=1..n, szigetkezdet(i) és keresvége(i).van, i) és
    (van,maxind,)=FELTMAX(i=1..db, táv(kezdetek[i]), igaz) és
    van -> k=kezdetek[maxind] és v=keresvége(k).ind
```

1. Válogassuk ki azokat a szigetkezdeteket, amelyeknek van vége is!
2. Másolással határozzuk meg hozzájuk a szigetvégeket! (összevonható)
3. Másolással határozzuk meg a távolságokat! (összevonható)
4. Határozzuk meg a legnagyobb távolságot, ha van!

1c. változat:



Specifikáció:

```
Be: n∈N, mag∈N[1..n]
Ki: van∈L, k∈N, v∈N
Fv: szigetkezdet:N->L,
    szigetkezdet(i)={hamis, ha i=1;
                     mag[i-1]=0 és mag[i]>0 egyébként}
Fv: szigetvég:N->L,
    szigetvég(i)={hamis, ha i=n;
                  mag[i+1]=0 és mag[i]>0 egyébként}
Fv: keresvége:N->L x N,
    keresvége(i)=KERES(j=i..n, szigetvég(j))
Fv: táv:N->N, táv(i)=keresvége(i).ind - i
Ef: n>=3 és mag[1]>0 és mag[n]>0
Uf: (van,k,)=FELTMAX(i=1..n, táv(i), szigetkezdet(i) és keresvége(i).van)
 és van -> v=keresvége(k).ind
```

1. Válogassuk ki azokat a szigetkezdeteket, amelyeknek van vége is! (összevonható)
2. Másolással határozzuk meg hozzájuk a szigetvégeket! (összevonható)
3. Másolással határozzuk meg a távolságokat! (összevonható)
4. Határozzuk meg a legnagyobb távolságot, ha van!

Feltételes maximumkeresésben keresés

1c. változat

```
Fv: keresvége:N->L x N,
    keresvége(i)=KERES(j=i..n, szigetvég(j))
Fv: táv:N->N, táv(i)=keresvége(i).ind - i
Ef: n>=3 és mag[1]>0 és mag[n]>0
Uf: (van,k,)=FELTMAX(i=1..n, táv(i), szigetkezdet(i) és keresvége(i).van)
 és van -> v=keresvége(k).ind
```

Feltételes maximumkeresés

maxind	~	k
e..u	~	1..n
f(i)	~	táv(i)
T(i)	~	szigetkezdet(i) és keresvége(i).van

Keresés (keresvége)

i	~	j
e..u	~	i..n
T(i)	~	szigetvég(j)

1c. változat

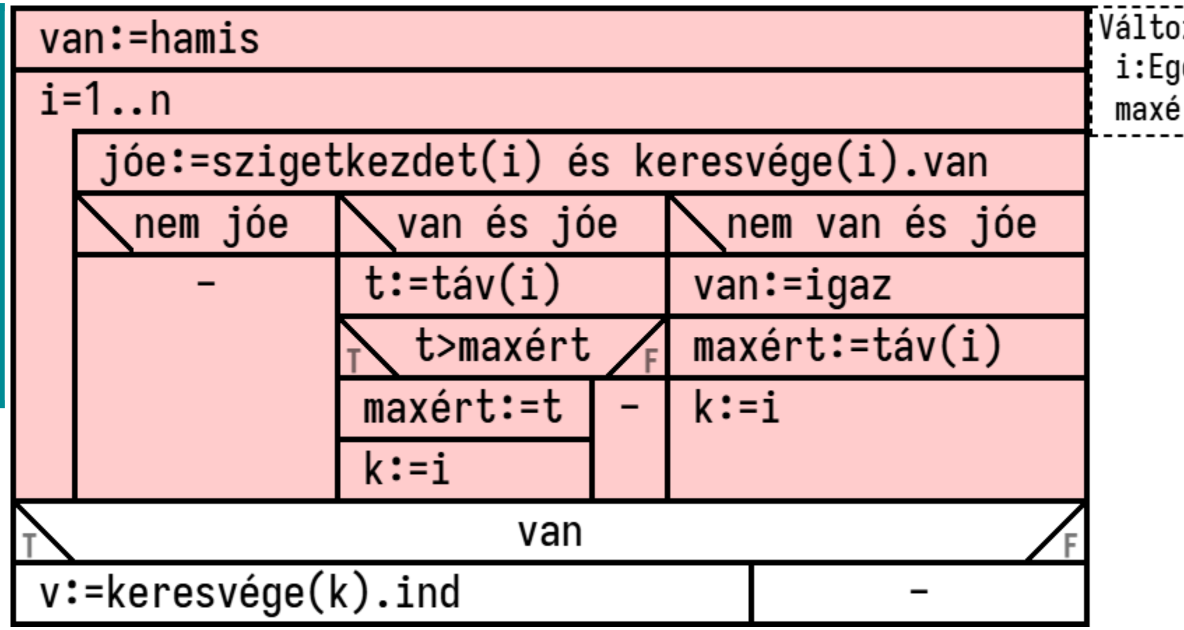
Feltételes maximumkeresés

maxind ~ k

e..u ~ 1..n

f(i) ~ táv(i)

T(i) ~ szigetkezdet(i) és keresvége(i).van

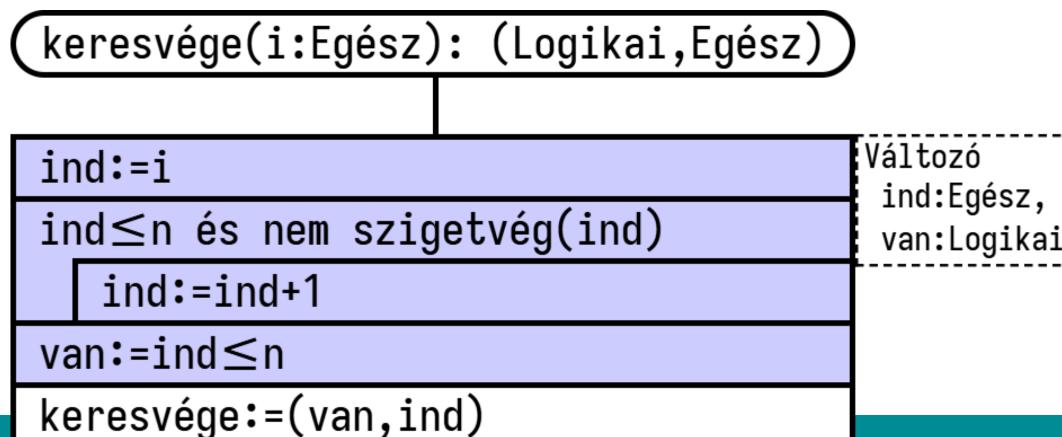


Keresés (keresvége)

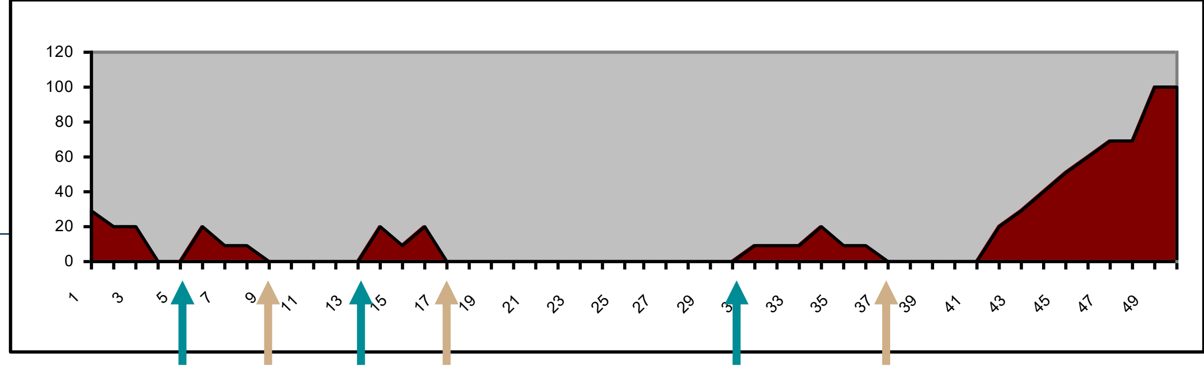
i ~ j

e..u ~ i..n

T(i) ~ szigetvég(j)



2. változat:



Specifikáció:

Be: n∈N, mag∈N[1..n]

Sa: db∈N, határok∈N[1..db], maxind∈N

Ki: van∈L, k∈N, v∈N

Fv: szigethatár:N->L,
    szigethatár(i)=
      (mag[i]>0 és mag[i+1]=0) vagy
      (mag[i]=0 és mag[i+1]>0)

Ef: n>=3 és mag[1]>0 és mag[n]>0

Uf: (db,határok)=**KIVÁLOGAT**(i=1..n-1, szigethatár(i), i) és
    (van,maxind,)=**FELTMAX**(i=1..(db-2) div 2,
                                    határok[2*i+1]-határok[2*i], igaz) és
    van -> (k=határok[2*maxind]+1 és v=határok[2*maxind+1])

1. Válogassuk ki a szigethatárokat (kezdet és vég is határ)!
2. Másolással határozzuk meg a távolságokat! (összevonható)
3. Határozzuk meg a legnagyobb távolságot, ha van!

Nem összevonható: a szomszédos elemek vizsgálatát a segédadat segíti. E nélkül az i. pontból mindig keresni kellene.

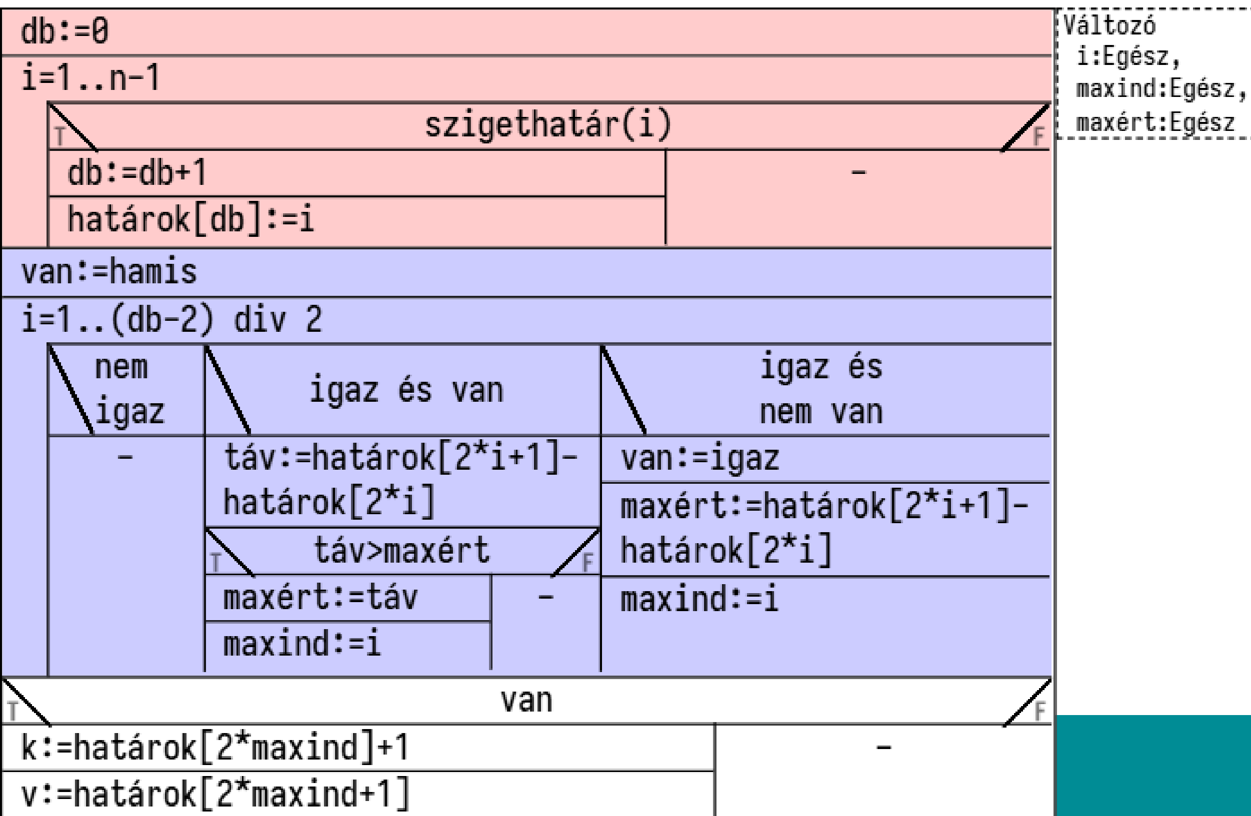
2. változat

Uf: (db,határok)=**KIVÁLOGAT**(i=1..n-1, szigethatár(i), i) és

(van,maxind,)=**FELTMAX**(i=1..(db-2) div 2,

határok[2*i+1]-határok[2*i], igaz) és

van -> (k=határok[2*maxind]+1 és v=határok[2*maxind+1])



Kiválogatás

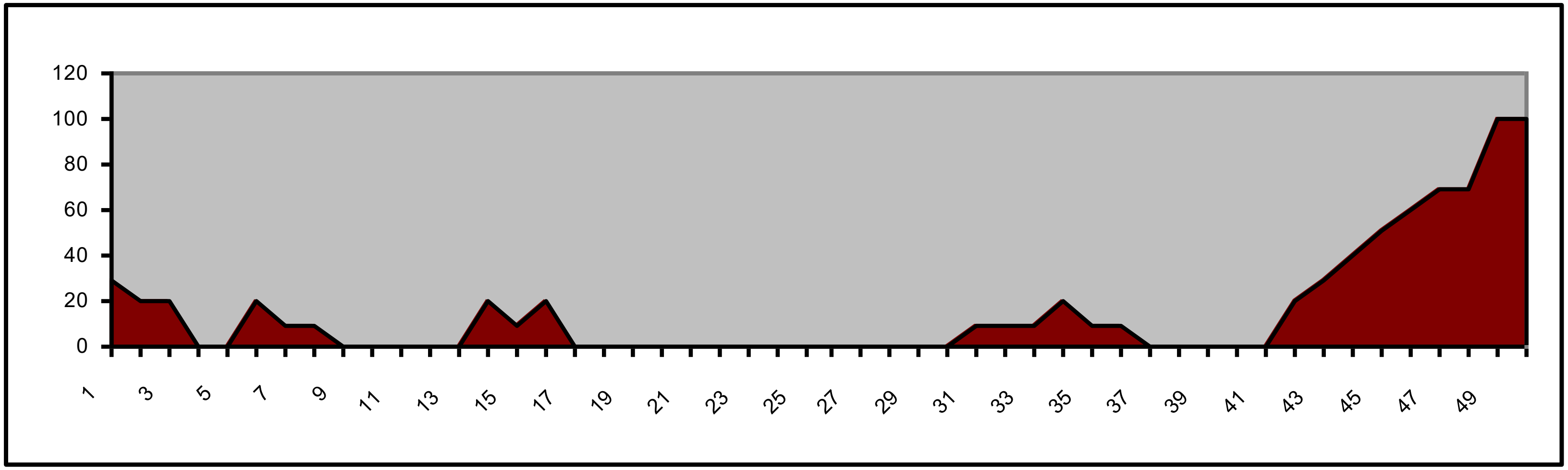
y	~	határok
e..u	~	1..n-1
T(i)	~	szigethatár(i)
f(i)	~	i

Feltételes maximumkeresés

e..u	~	1..(db-2) div 2
f(i)	~	határok[2*i+1]-határok[2*i]
T(i)	~	igaz

További változatok

- Mi és hogyan változik, ha
  - azt is szigetnek tekintem, ami esetleg a szélére esik? (ami eddig a kontinens volt)
  - a legrövidebb szigetet keresem?



Intervallum

Feladat:

Egy polgárőrség nyilvántartásában n őr van, és tudjuk minden őrről, hogy m napból mely naptól mely napig volt szolgálatban. Melyik volt az a leghosszabb időszak, amíg védve volt a helység?

őr	tól	ig	nap=	1	2	3	4	5	6	7	8	9	10
1	1	3		x	x	x							
2	2	2			x								
3	5	8						x	x	x	x		
4	7	8								x	x		
n=5	8	9									x	x	
			db=	1	2	1	0	1	1	2	3	1	0

Intervallum

őr	tól	ig
1	1	3
2	2	2
3	5	8
4	7	8
n=5	8	9

Feladat: leghosszabb időszak, amíg védve

nap=	1	2	3	4	5	6	7	8	9	10
	x	x	x							
		x								
					x	x	x	x		
							x	x		
								x	x	
db=	1	2	1	0	1	1	2	3	1	0

Specifikáció:

```
Be: m∈N, n∈N, őrök∈Intervallum[1..n],
    Intervallum=(tól:N x ig:N)
Sa: db∈N[1..m]
Ki: maxind∈N
Fv: hányőr:N->N,
    hányőr(nap)=DARAB(i=1..n, őrök[i].tól<=nap<=őrök[i].ig)
Ef: m>0 és ∀i∈[1..n]:(őrök[i].tól>=1 és őrök[i].ig<=m)
Uf: ∀nap∈[1..m]:(db[nap]=hányőr(nap)) és
    ...
```

Kezdődhet a leghosszabb „sziget” meghatározása!