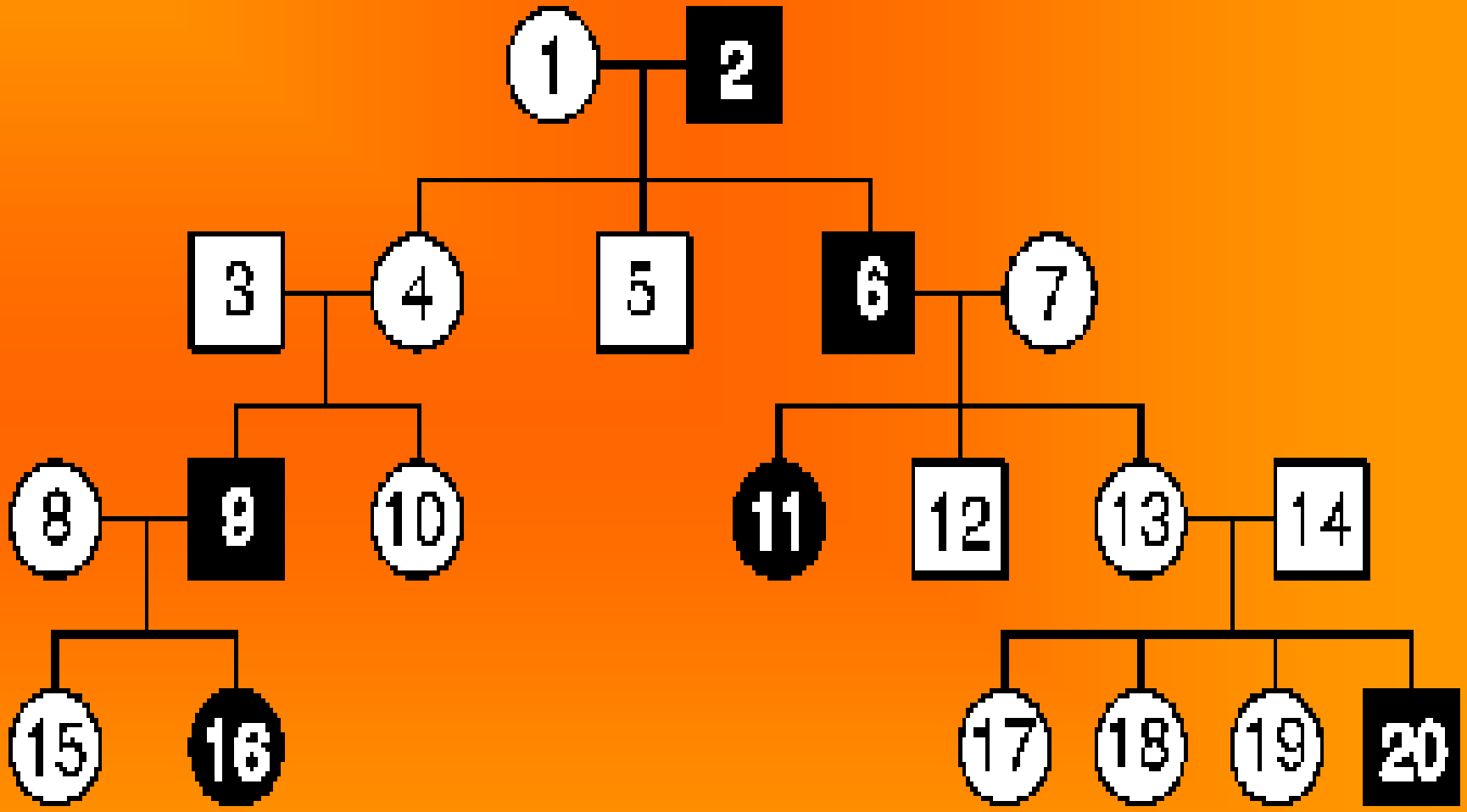
Parallel algorithm for sorting animal pedigrees

Maciej Gierdziewicz

The Faculty of Animal Sciences
Agricultural University in Cracow

Animal pedigree (example)



Relationships among animals

In many analyses of animal genotype, including the use of mixed model equations (MME) to estimate breeding values, there is a need to account for relationships among individuals.

Mixed Model Equations (MME) (I)

$$y = X\beta + Zu + e$$

Mixed Model Equations (MME) (II)

$$\begin{bmatrix} X'X & X'Z \\ Z'X & Z'Z + A^{-1}\lambda \end{bmatrix} \begin{bmatrix} \hat{\beta} \\ \hat{u} \end{bmatrix} = \begin{bmatrix} X'y \\ Z'y \end{bmatrix}$$

$$\lambda = \sigma_e^2 / \sigma_a^2$$

Relationships among animals - example (1)

WOMBAT

Version 1.0

A program for Mixed Model Analyses by Restricted Maximum Likelihood

USER NOTES

Karin Meyer
Animal Genetics and Breeding Unit,
University of New England
Armidale, NSW 2351,
AUSTRALIA

Relationships among animals - example (2)

[...]

6.2 Pedigree File

[...]

Coding

4. All animals must have a numerically higher code than either of their parents.

Unknown parents are to be coded as “0”.

[...]

Accounting for relationships - another example

Boldman K.G., L.A. Kriese, L.D. Van Vleck, L.A. Van Tassel, S.D.Kachman, 1993. A manual for use of MTDFREML, a set of programs to obtain estimates of variances and covariances. USDA-ARS, Clay Center, Nebraska, USA.

Breeding Value Estimation (BVE)

Incorrectly calculated relationship coefficients may lead to bias in estimation of genetic values.

Sorting animal pedigrees (1)

The chronological order of pedigrees is easy to achieve when they include birth dates.

1	*0*	*0*	*1990*
2	*0*	*0*	*1991*
3	*0*	*0*	*1991*
4	*1*	*2*	*1995*
5	*1*	*3*	*1996*
6	*4*	*5*	*2000*

...

Sorting animal pedigrees (2)

In extreme cases, when (almost) no birth dates are present, the inference about the order of the animals must be made by comparing at least once each pair of individuals separately.

Simplest method to sort pedigrees (method I)

```
DO  i = 1 , n - 1
      DO  j = i + 1 , n
            rel = compare ( i , j )
            IF ( rel .eq. ">" ) CALL swap ( i , j )
      END DO
END DO
```

Example of sorting pedigrees: „pyramid" algorithm (method II)

Zhang Z., C. Li, R.J. Todhunter, G. Lust, L. Goonewardene, Z. Wang, 2009. An Algorithm to Sort Complex Pedigrees Chronologically without Birthdates. Journal of Animal and Veterinary Advances 8 (1): 177-182.

„Pyramid” algorithm (method II) (continued)

1	4	12	1	2	2	2	4 3
2	11	13	1	2	2	2	4 9
3	0	0	1	2	3	4	3 4
4	3	9	1	2	3	3	3 11
5	14	15	1	2	2	2	3 12
6	5	10	1	1	1	1	3 13
7	6	8	0	0	0	0	3 14
8	2	1	1	1	1	1	3 15
9	0	0	1	2	3	4	2 1
10	11	13	1	2	2	2	2 2
11	3	9	1	2	3	3	2 5
12	0	0	1	2	3	3	2 10
13	0	0	1	2	3	3	1 6
14	0	0	1	2	3	3	1 8
15	3	9	1	2	3	3	0 7

„Pyramid” algorithm (method II) (continued)

1	4	12	1	2	2	2	4 3
2	11	13	1	2	2	2	4 9
3	0	0	1	2	3	4	3 4
4	3	9	1	2	3	3	3 11
5	14	15	1	2	2	2	3 12
6	5	10	1	1	1	1	3 13
7	6	8	0	0	0	0	3 14
8	2	1	1	1	1	1	3 15
9	0	0	1	2	3	4	2 1
10	11	13	1	2	2	2	2 2
11	3	9	1	2	3	3	2 5
12	0	0	1	2	3	3	2 10
13	0	0	1	2	3	3	1 6
14	0	0	1	2	3	3	1 8
15	3	9	1	2	3	3	0 7

„Pyramid" algorithm (method II) (continued)

1	4	12	1	2	2	2	4 3
2	11	13	1	2	2	2	4 9
3	0	0	1	2	3	4	3 4
4	3	9	1	2	3	3	3 11
5	14	15	1	2	2	2	3 12
6	5	10	1	1	1	1	3 13
7	6	8	0	0	0	0	3 14
8	2	1	1	1	1	1	3 15
9	0	0	1	2	3	4	2 1
10	11	13	1	2	2	2	2 2
11	3	9	1	2	3	3	2 5
12	0	0	1	2	3	3	2 10
13	0	0	1	2	3	3	1 6
14	0	0	1	2	3	3	1 8
15	3	9	1	2	3	3	0 7

„Pyramid” algorithm (method II) (continued)

1	4	12	1	2	2	2	4 3
2	11	13	1	2	2	2	4 9
3	0	0	1	2	3	4	3 4
4	3	9	1	2	3	3	3 11
5	14	15	1	2	2	2	3 12
6	5	10	1	1	1	1	3 13
7	6	8	0	0	0	0	3 14
8	2	1	1	1	1	1	3 15
9	0	0	1	2	3	4	2 1
10	11	13	1	2	2	2	2 2
11	3	9	1	2	3	3	2 5
12	0	0	1	2	3	3	2 10
13	0	0	1	2	3	3	1 6
14	0	0	1	2	3	3	1 8
15	3	9	1	2	3	3	0 7

„Pyramid" algorithm (method II) (continued)

1	4	12	1	2	2	2	4 3
2	11	13	1	2	2	2	4 9
3	0	0	1	2	3	4	3 4
4	3	9	1	2	3	3	3 11
5	14	15	1	2	2	2	3 12
6	5	10	1	1	1	1	3 13
7	6	8	0	0	0	0	3 14
8	2	1	1	1	1	1	3 15
9	0	0	1	2	3	4	2 1
10	11	13	1	2	2	2	2 2
11	3	9	1	2	3	3	2 5
12	0	0	1	2	3	3	2 10
13	0	0	1	2	3	3	1 6
14	0	0	1	2	3	3	1 8
15	3	9	1	2	3	3	0 7

The aim of the work

The aim of the work was to create a parallel version of the simple algorithm to compare and sort animal pedigrees without birth dates, basing on relationships between ancestors and progeny.

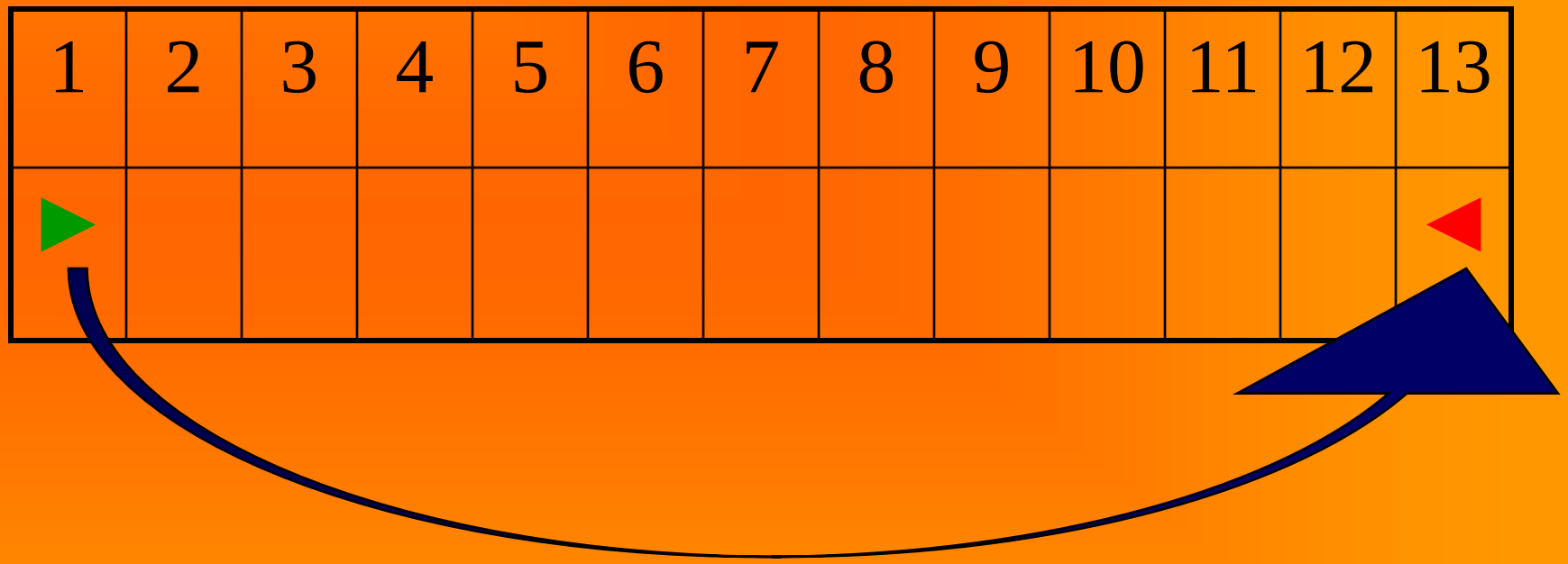
D A T A

63264 one-generation cattle pedigrees

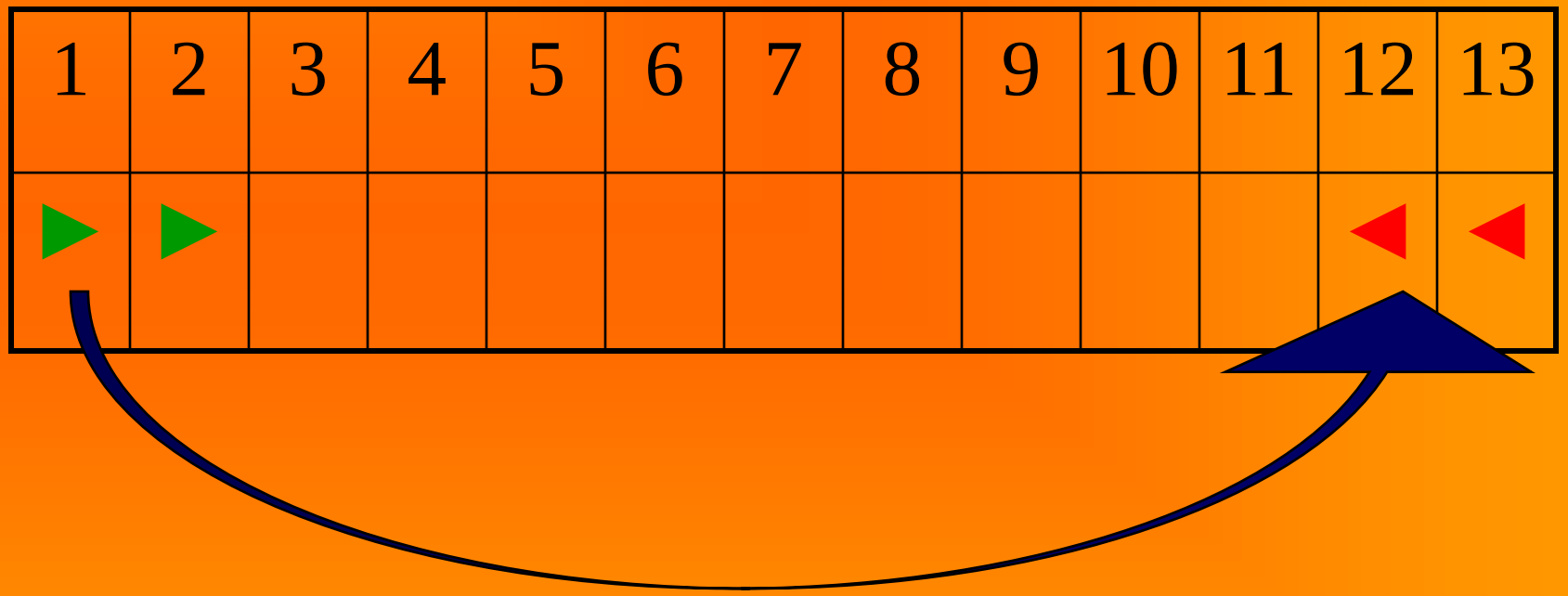
animal - sire - dam

(63264 animals found in pedigrees of Polish Black-and-White bulls born 1960-2000)

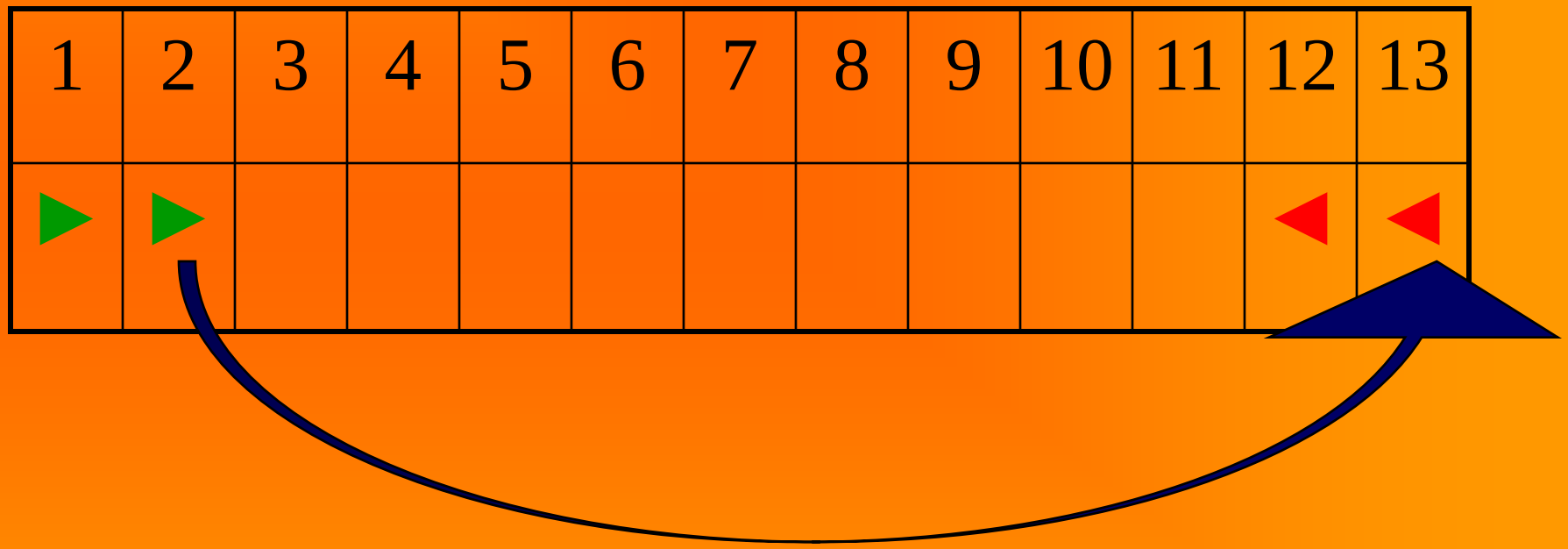
Parallelization of method I (distance = 12)



Parallelization of method I (distance = 11)

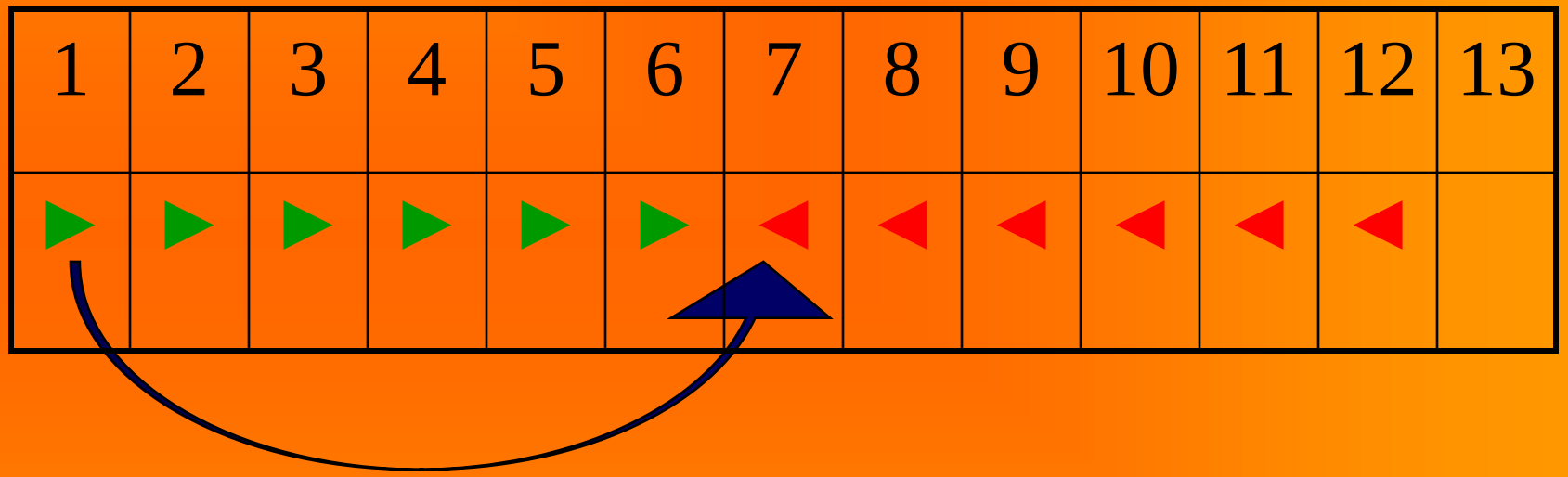


Parallelization of method I (distance = 11)

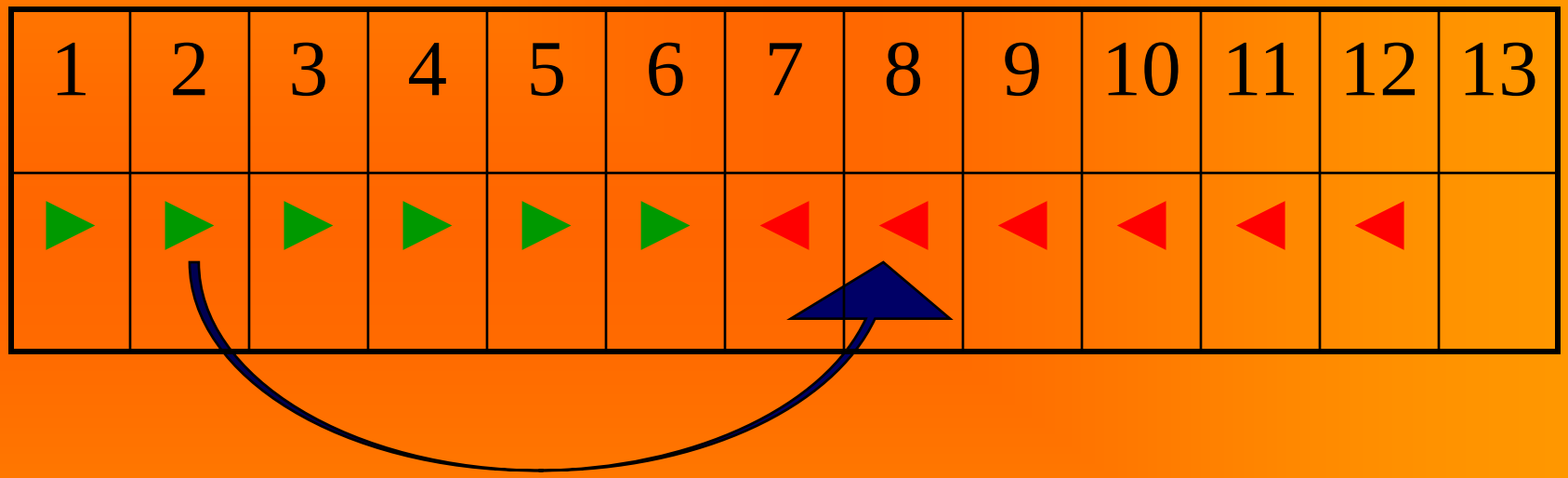


etc. (10 , 9 , ...)

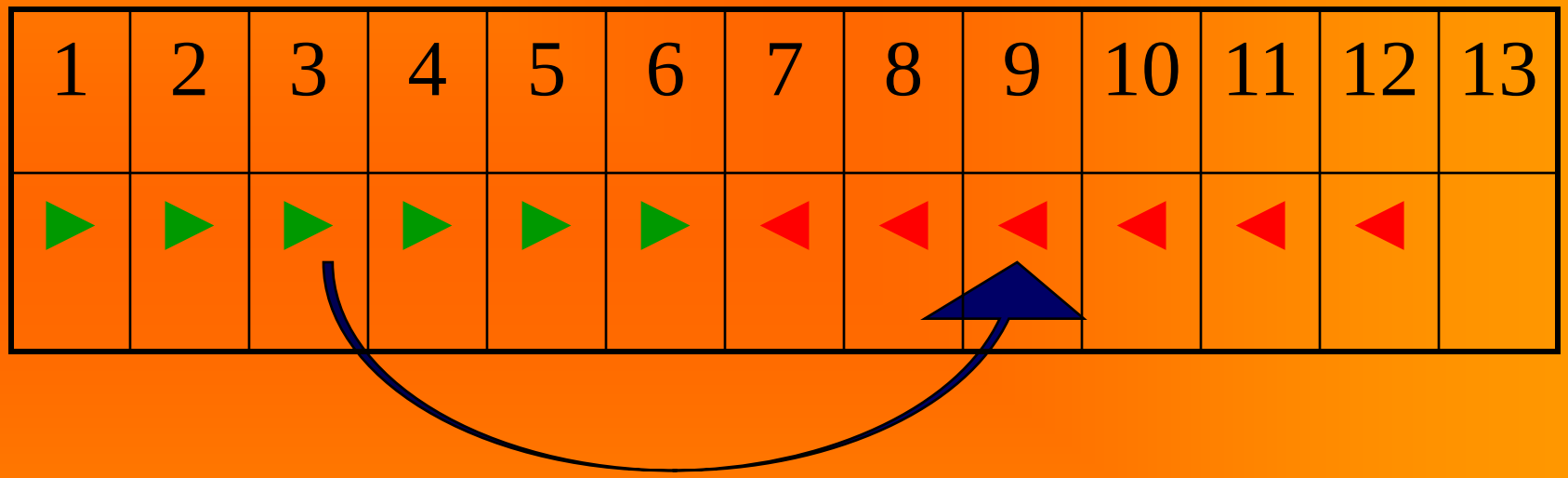
Parallelization of method I (distance = 6 , phase 1)



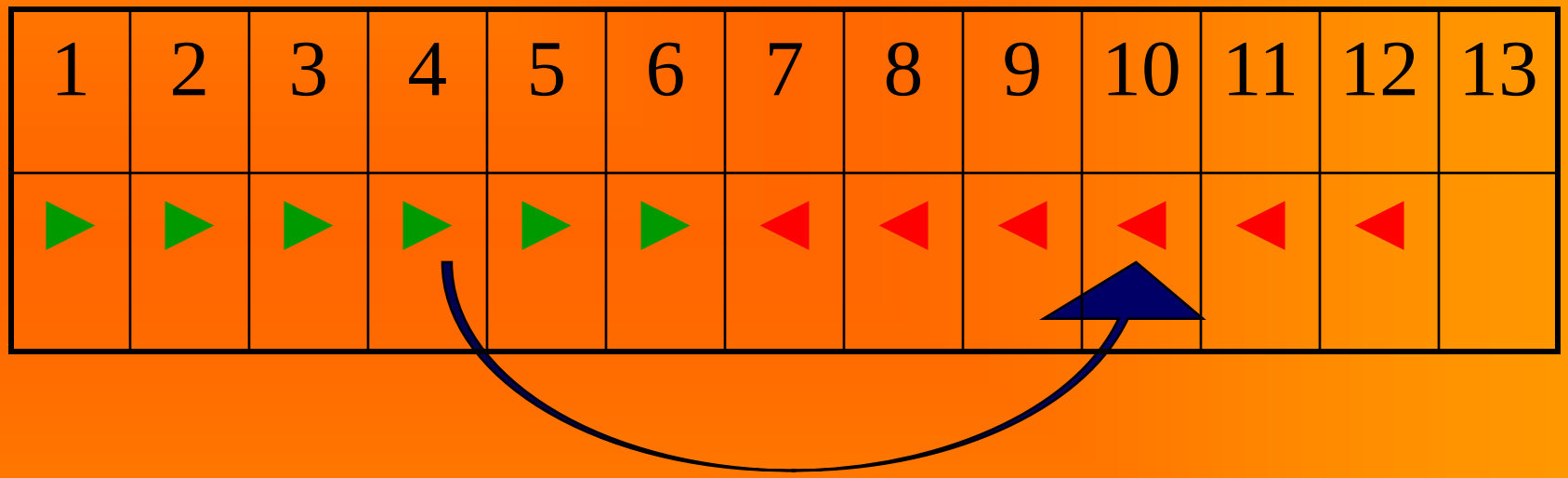
Parallelization of method I (distance = 6 , phase 1)



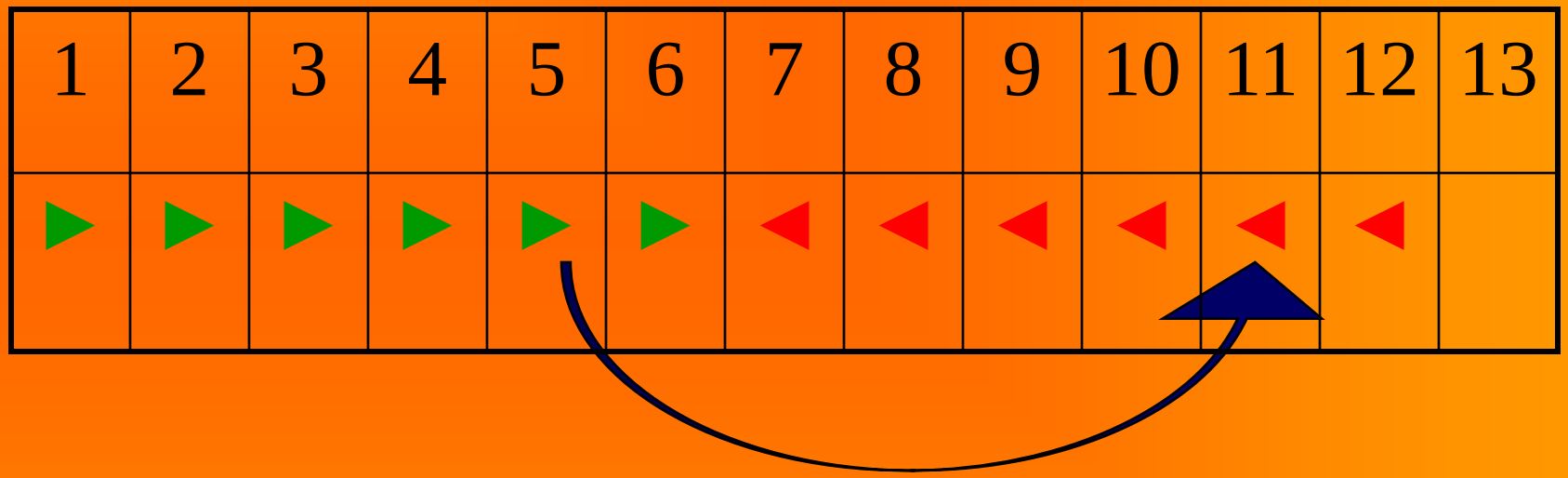
Parallelization of method I (distance = 6 , phase 1)



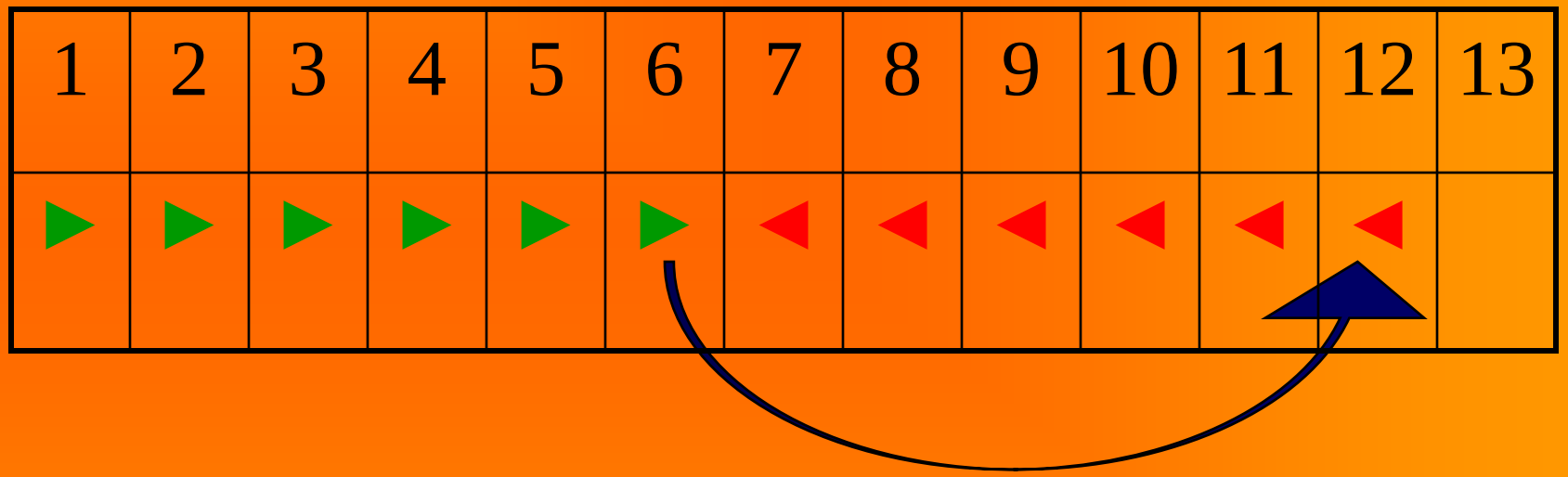
Parallelization of method I (distance = 6 , phase 1)



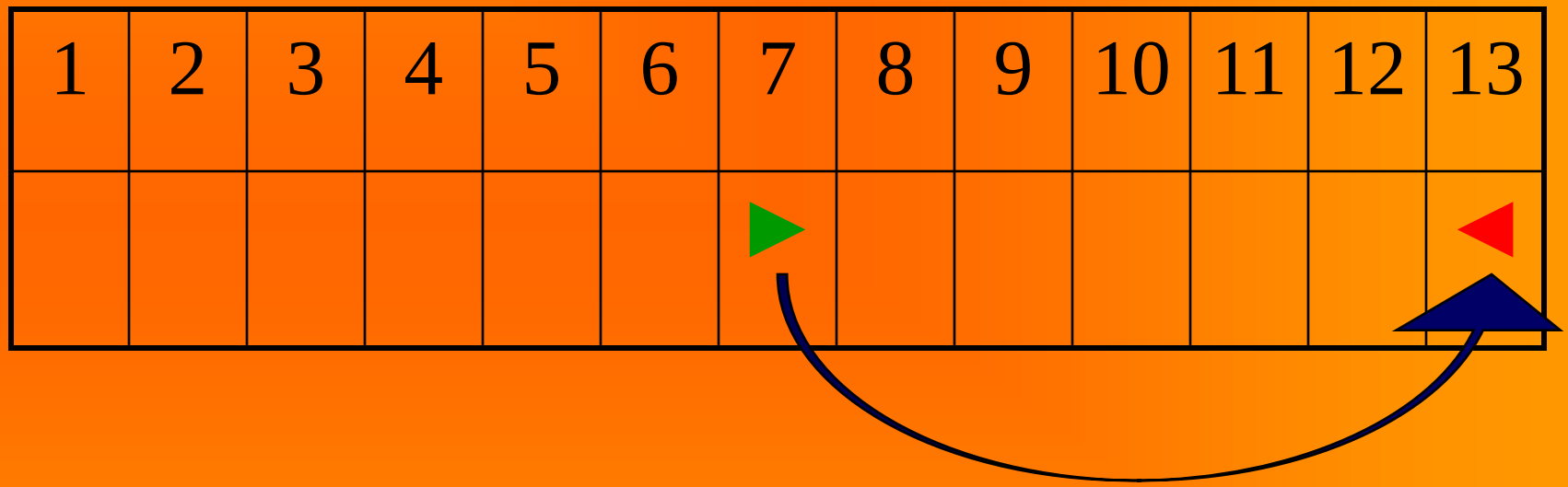
Parallelization of method I (distance = 6 , phase 1)



Parallelization of method I (distance = 6 , phase 1)

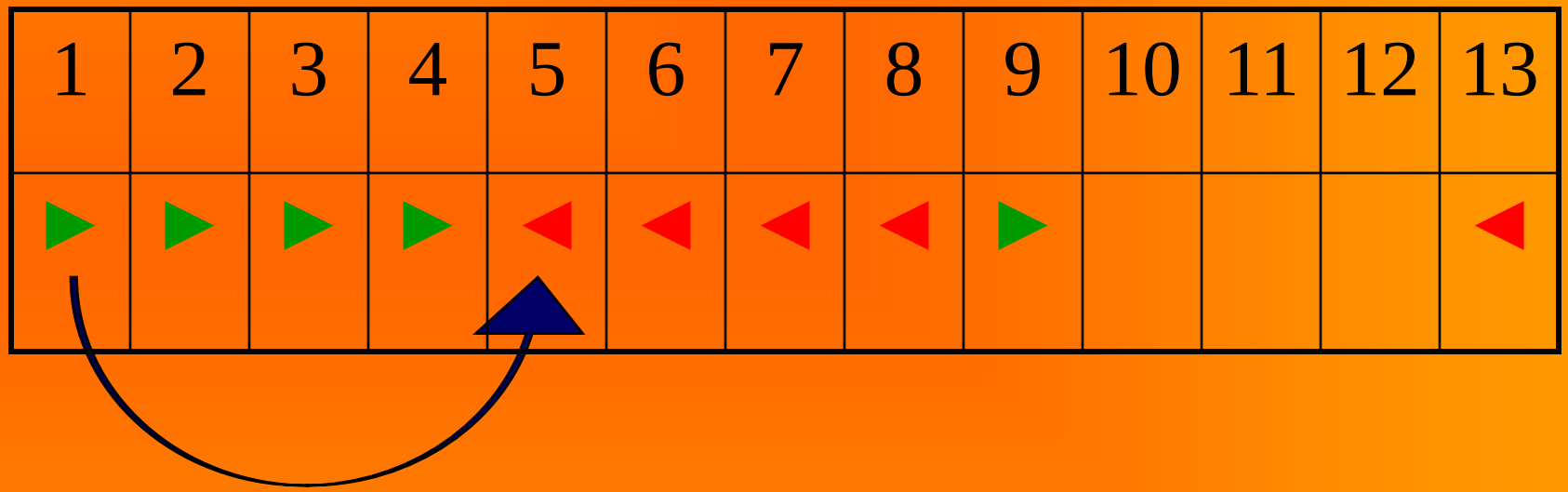


Parallelization of method I (distance = 6 , phase 2)

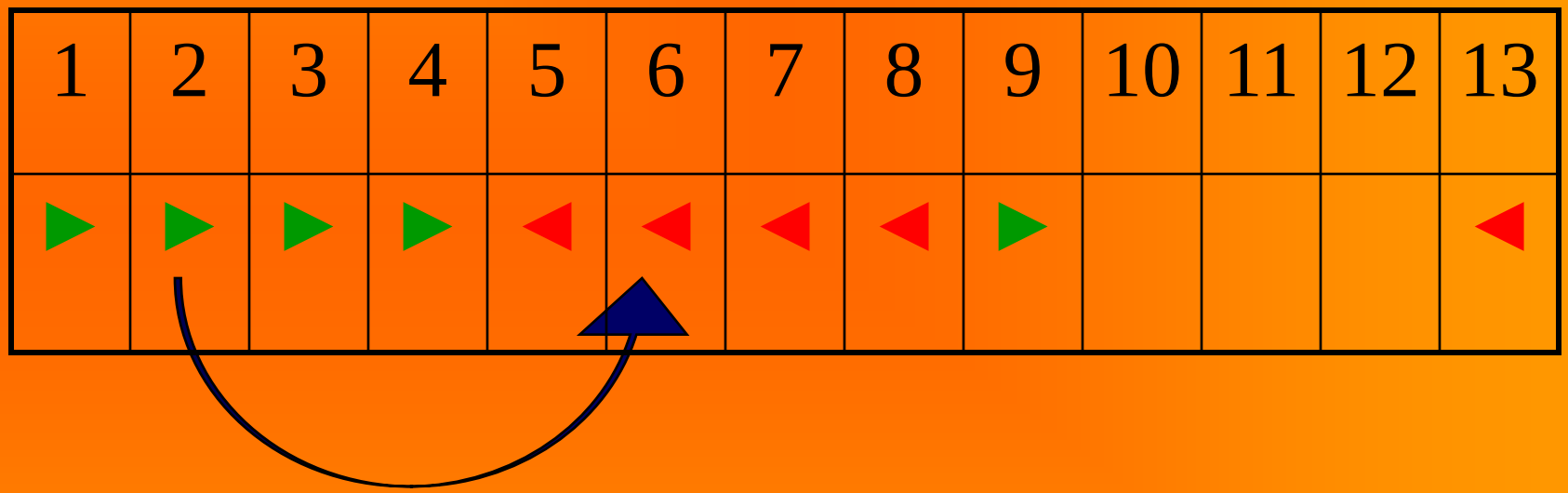


(distance = 5 : likewise)

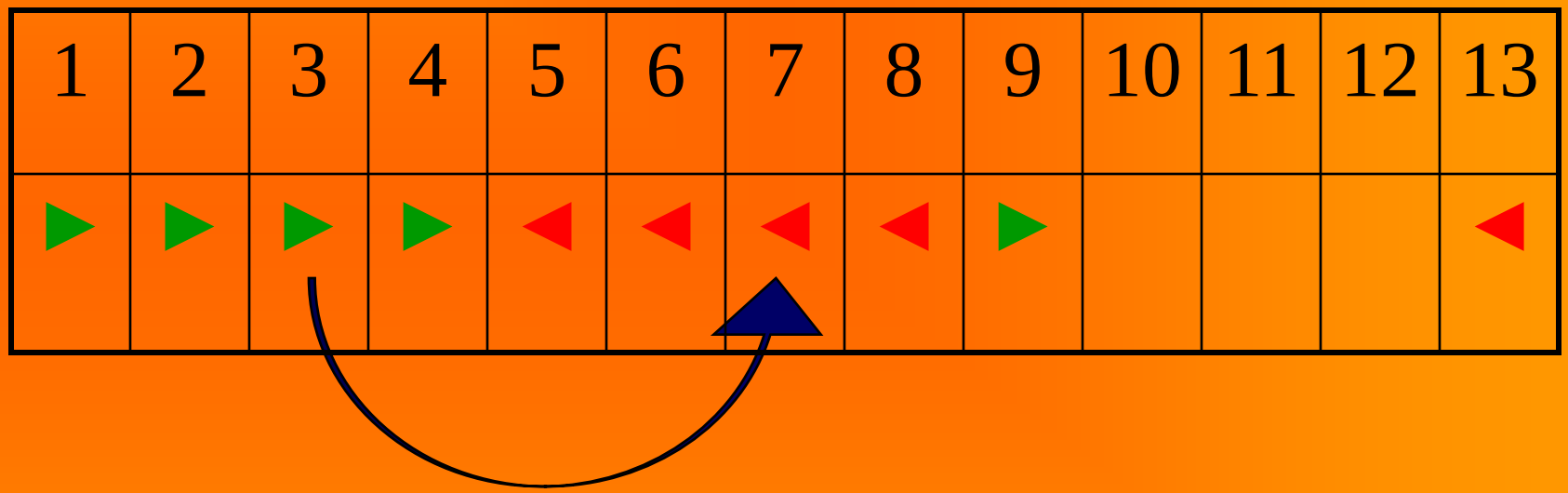
Parallelization of method I (distance = 4 , phase 1)



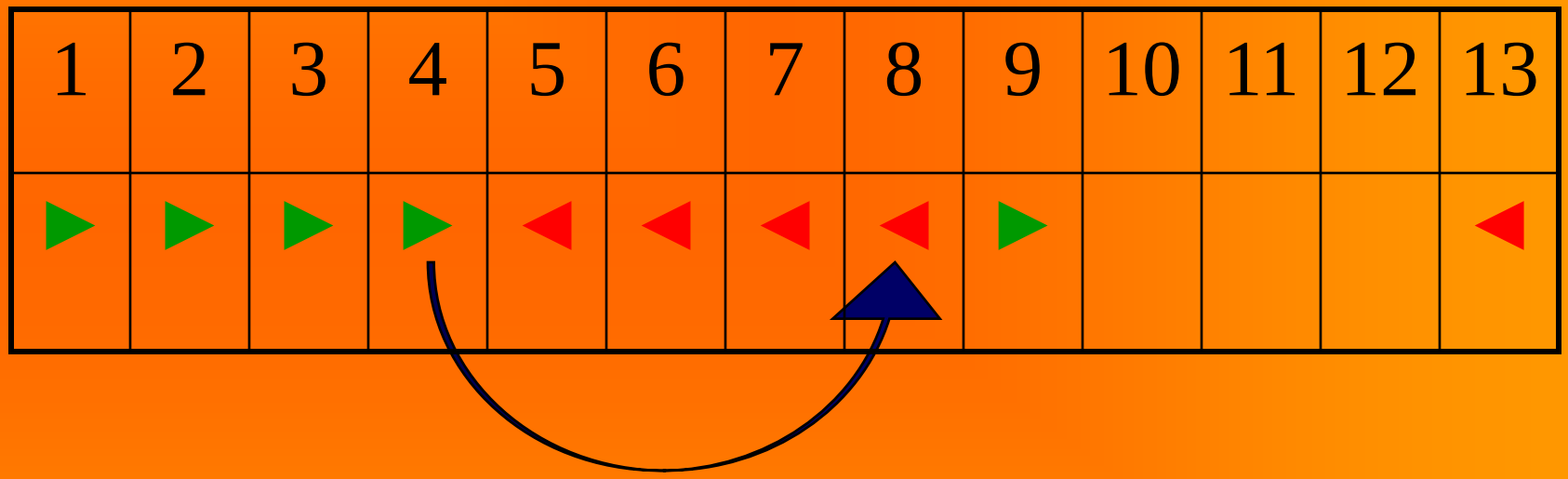
Parallelization of method I (distance = 4 , phase 1)



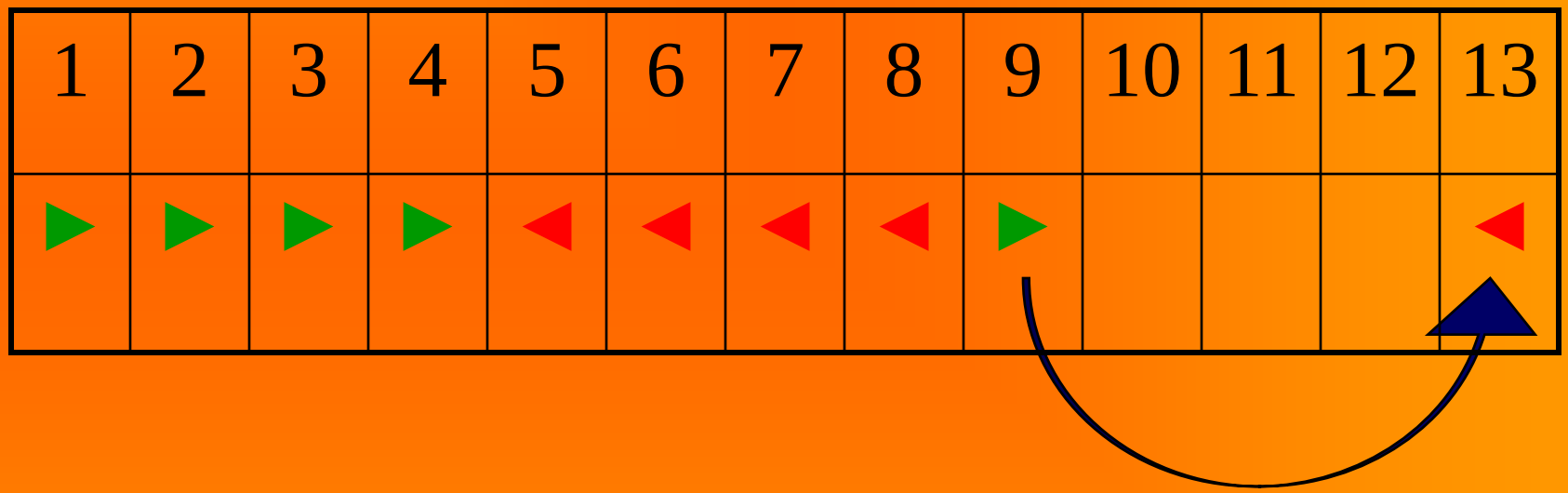
Parallelization of method I (distance = 4 , phase 1)



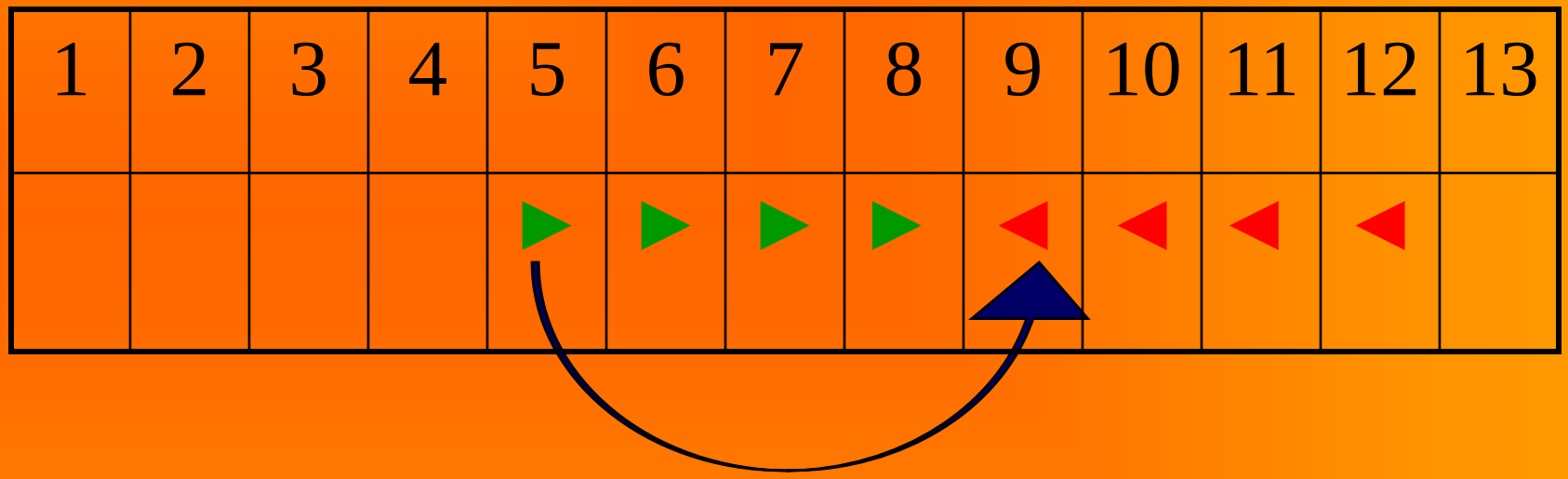
Parallelization of method I (distance = 4 , phase 1)



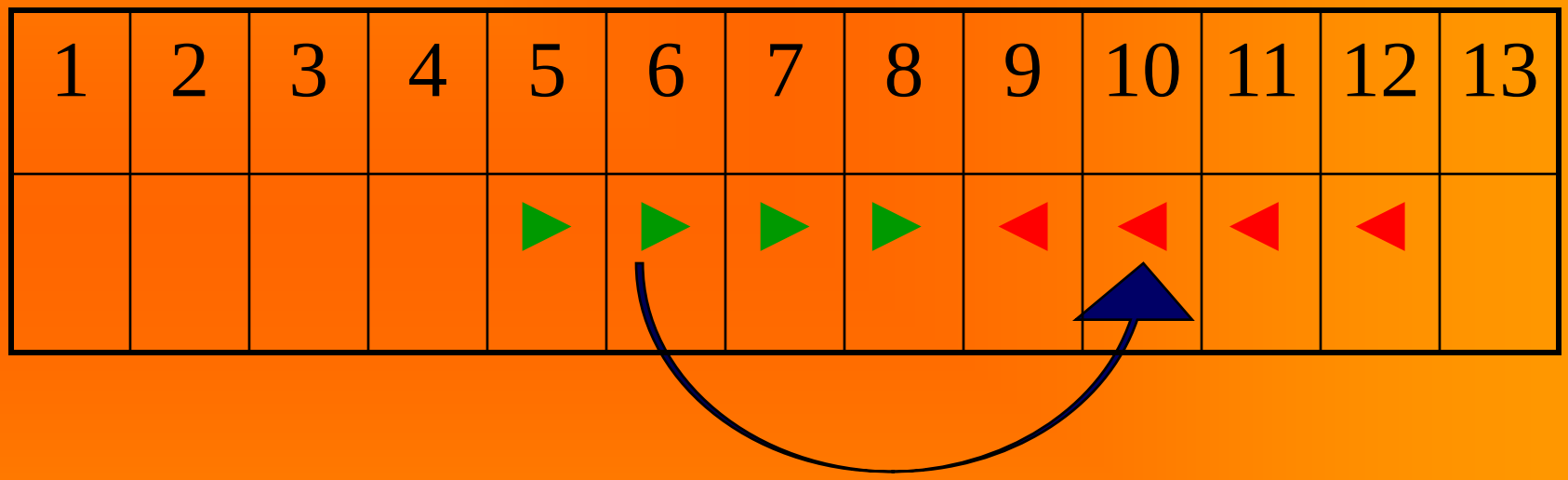
Parallelization of method I (distance = 4 , STILL phase 1)



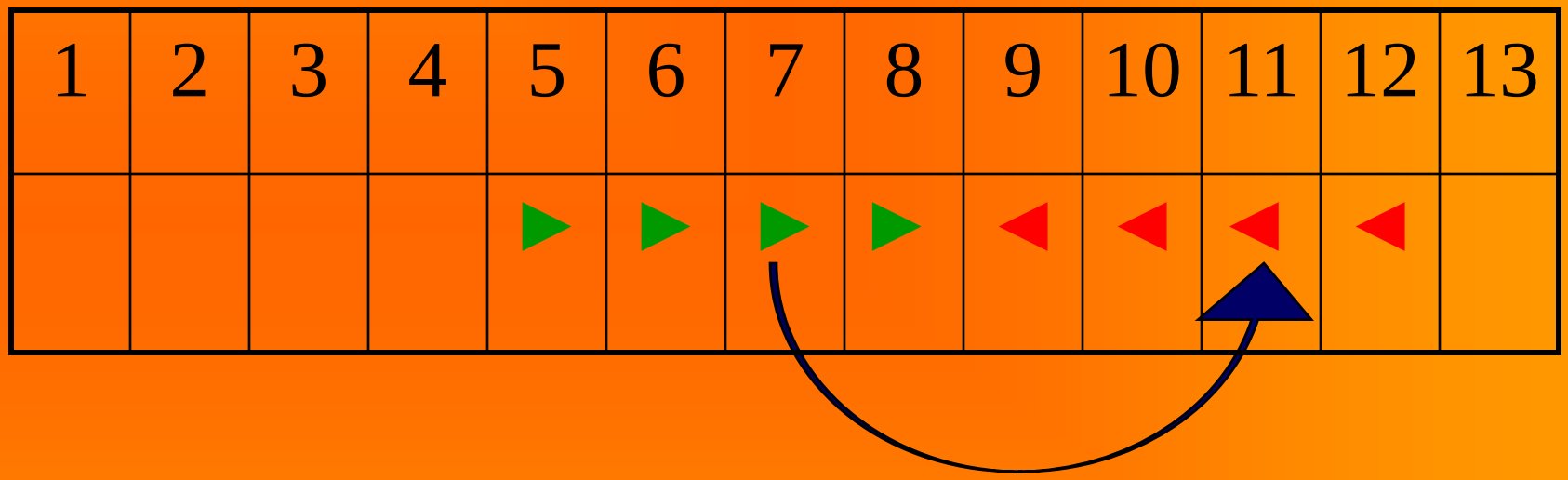
Parallelization of method I (distance = 4 , phase 2)



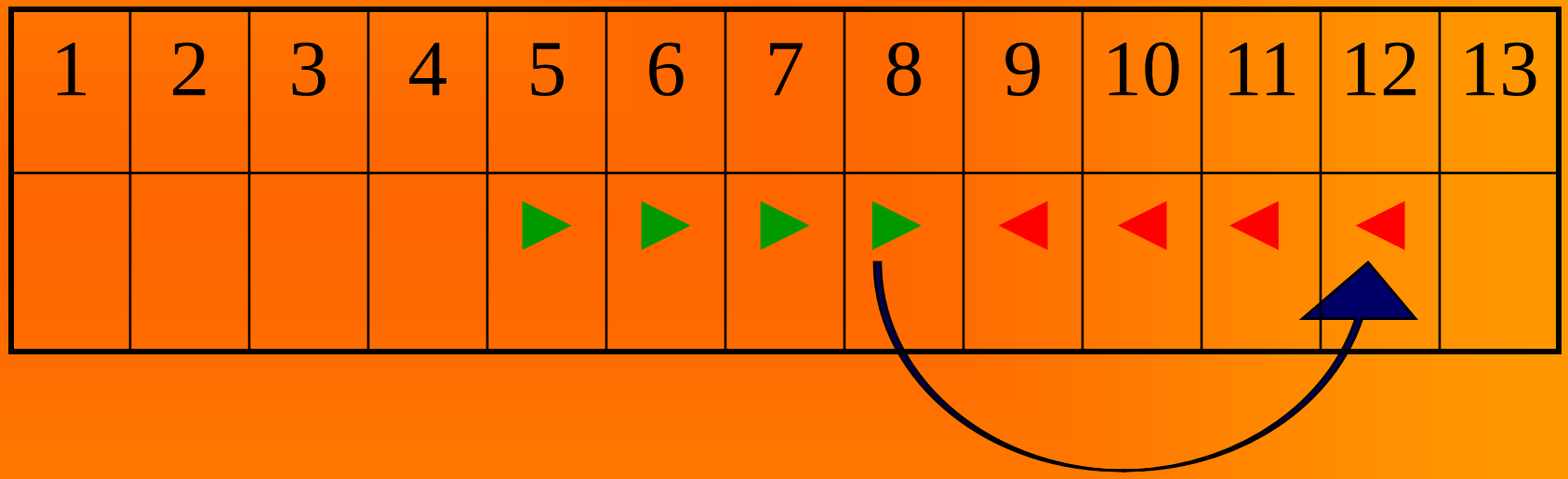
Parallelization of method I (distance = 4 , phase 2)



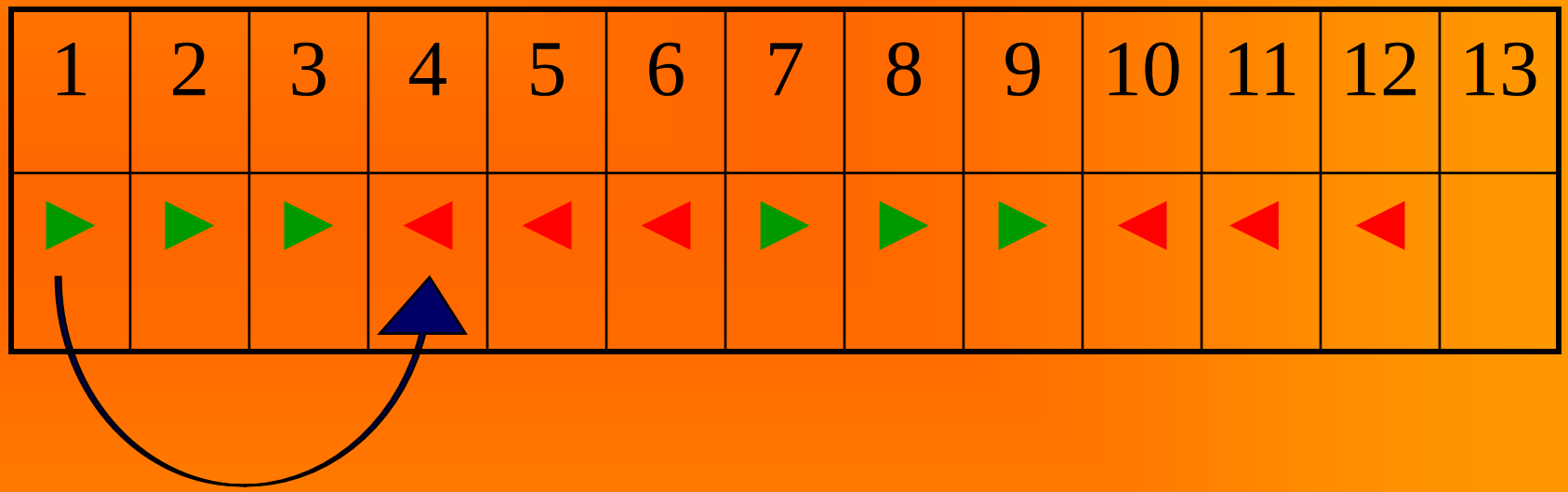
Parallelization of method I (distance = 4 , phase 2)



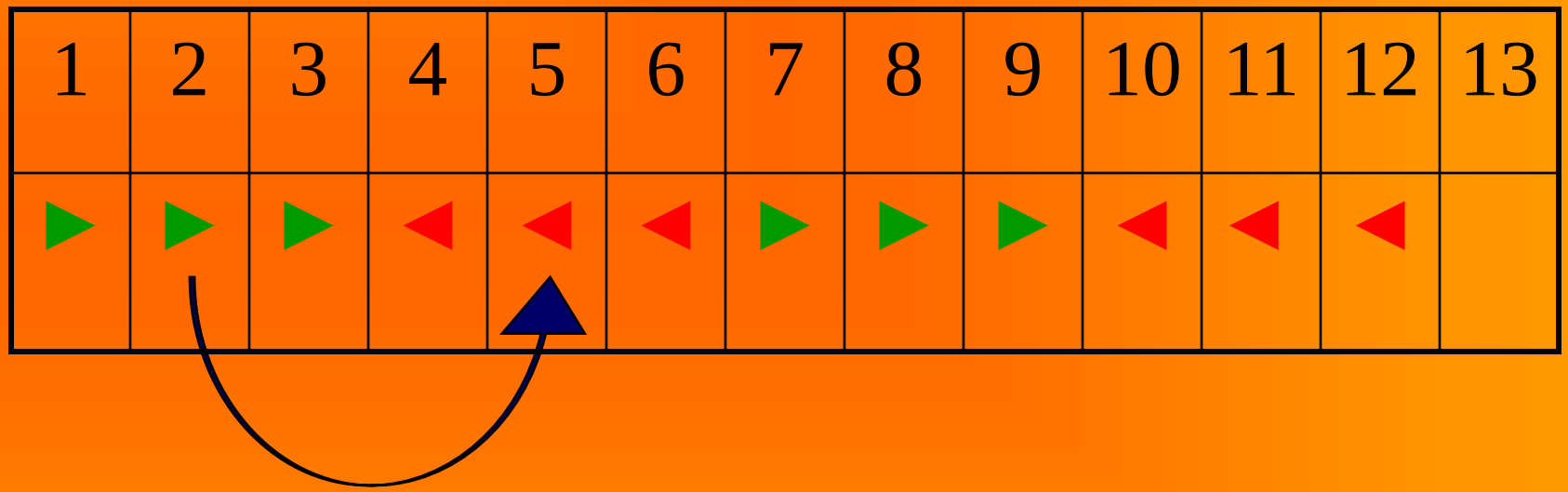
Parallelization of method I (distance = 4 , phase 2)



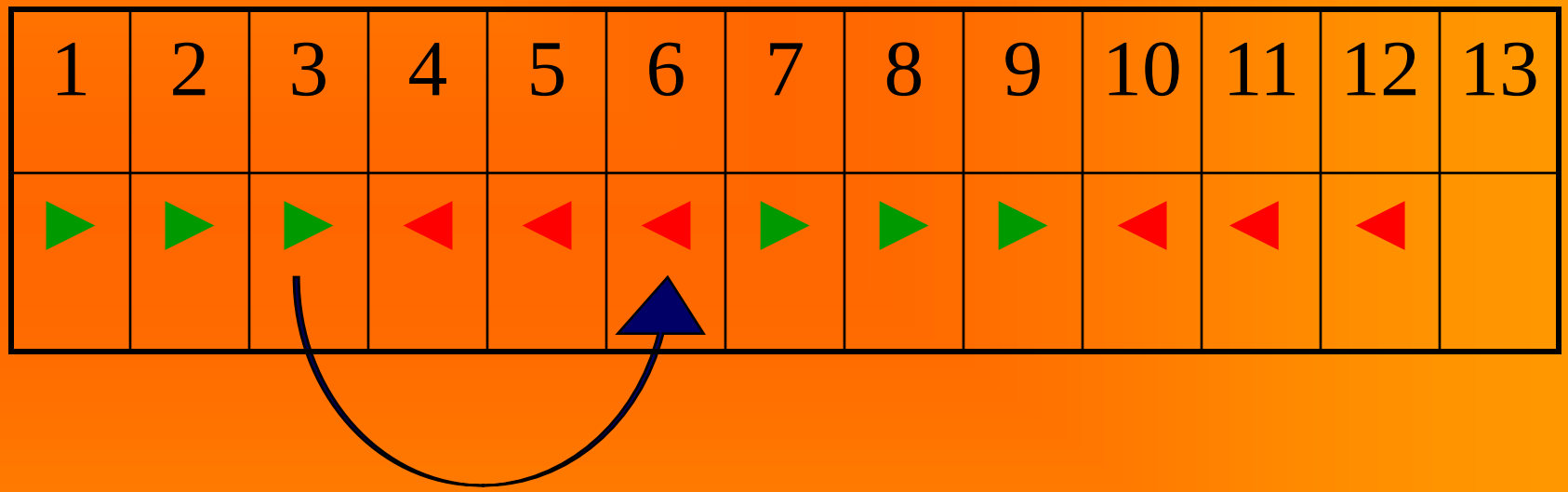
Parallelization of method I (distance = 3 , phase 1)



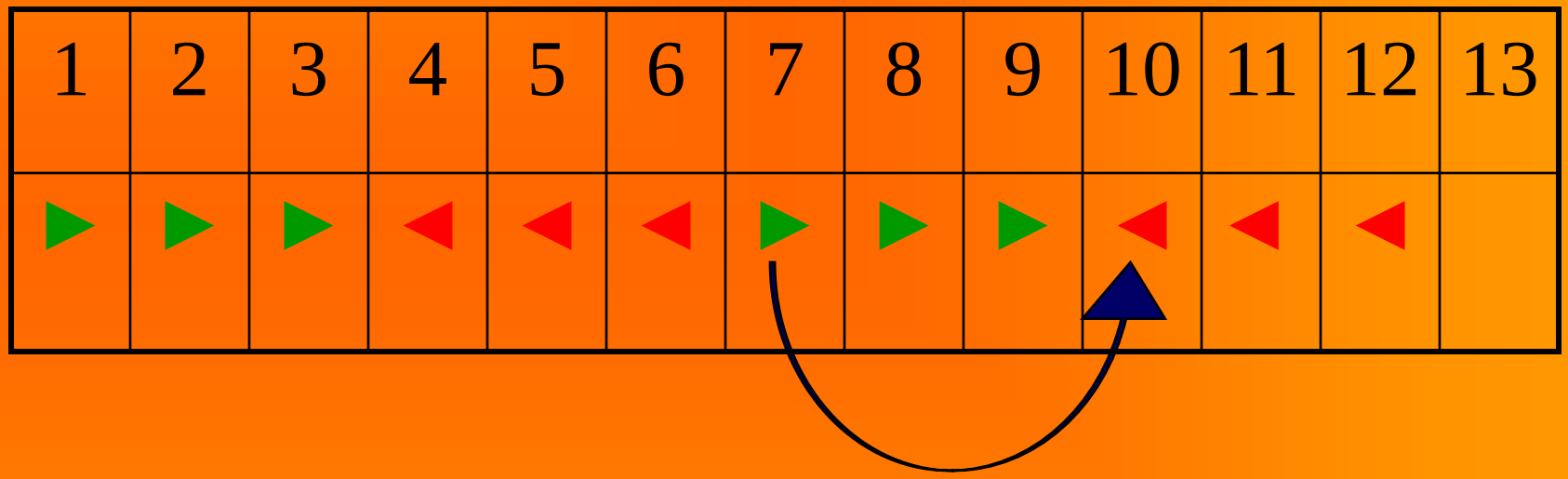
Parallelization of method I (distance = 3 , phase 1)



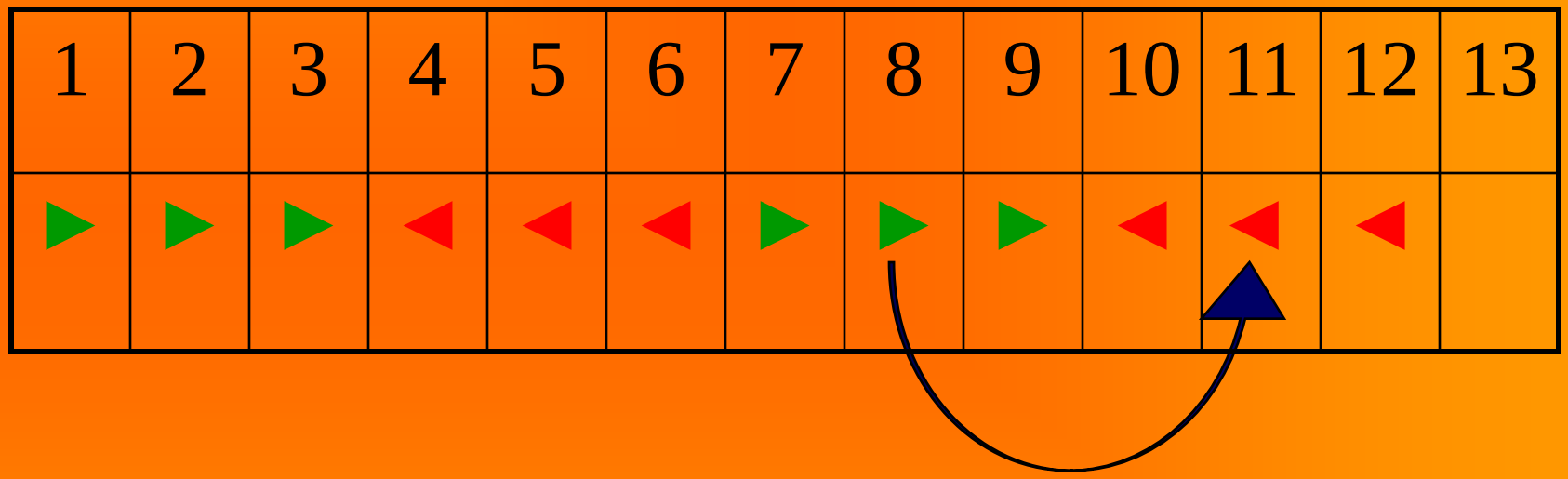
Parallelization of method I (distance = 3 , phase 1)



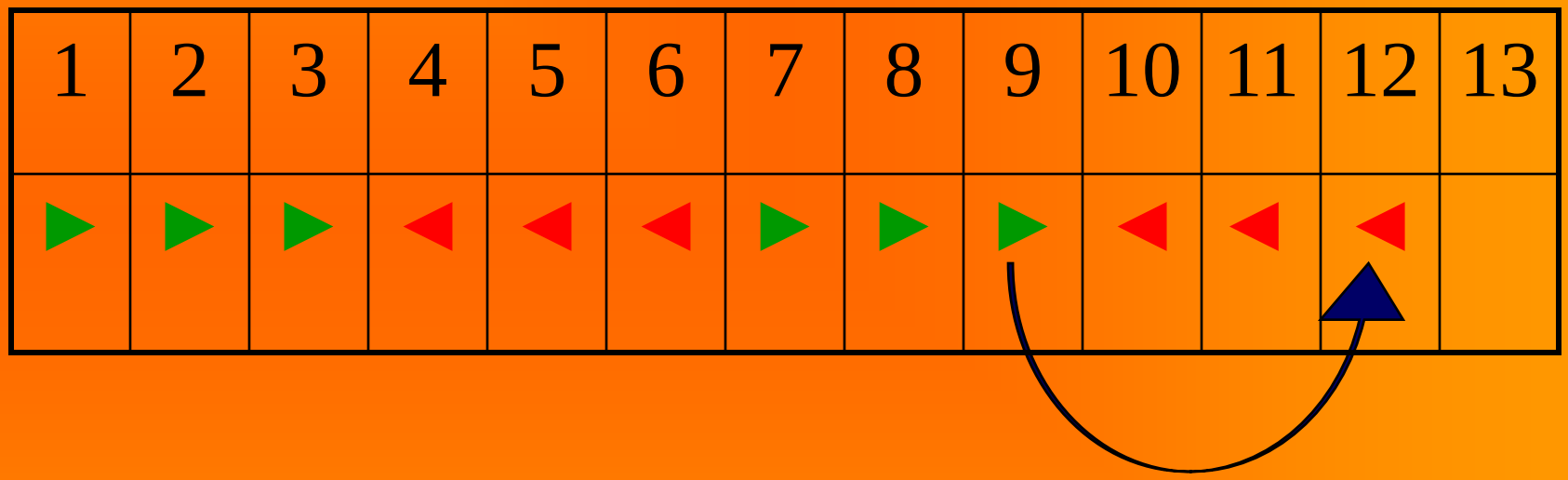
Parallelization of method I (distance = 3 , phase 1)



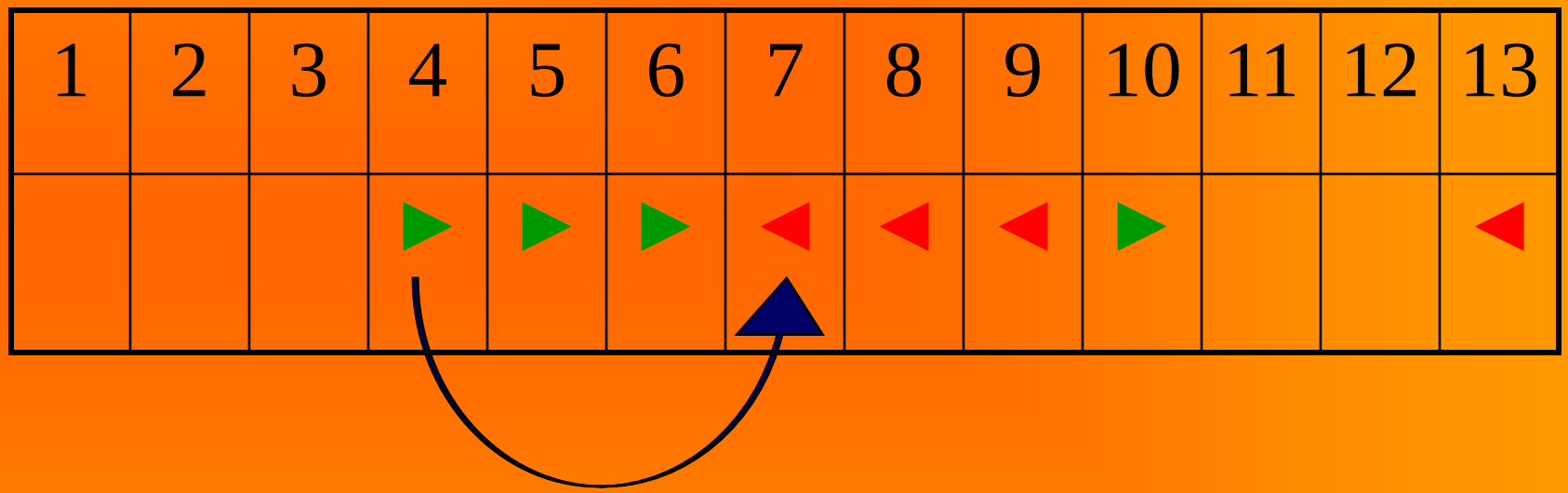
Parallelization of method I (distance = 3 , phase 1)



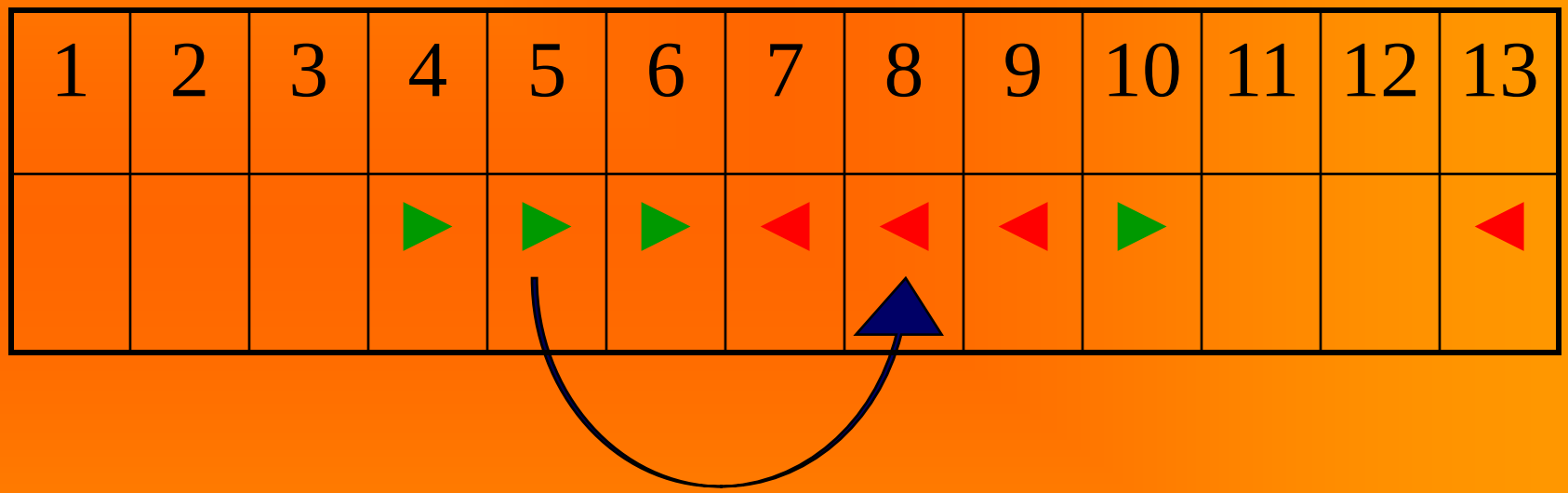
Parallelization of method I (distance = 3 , phase 1)



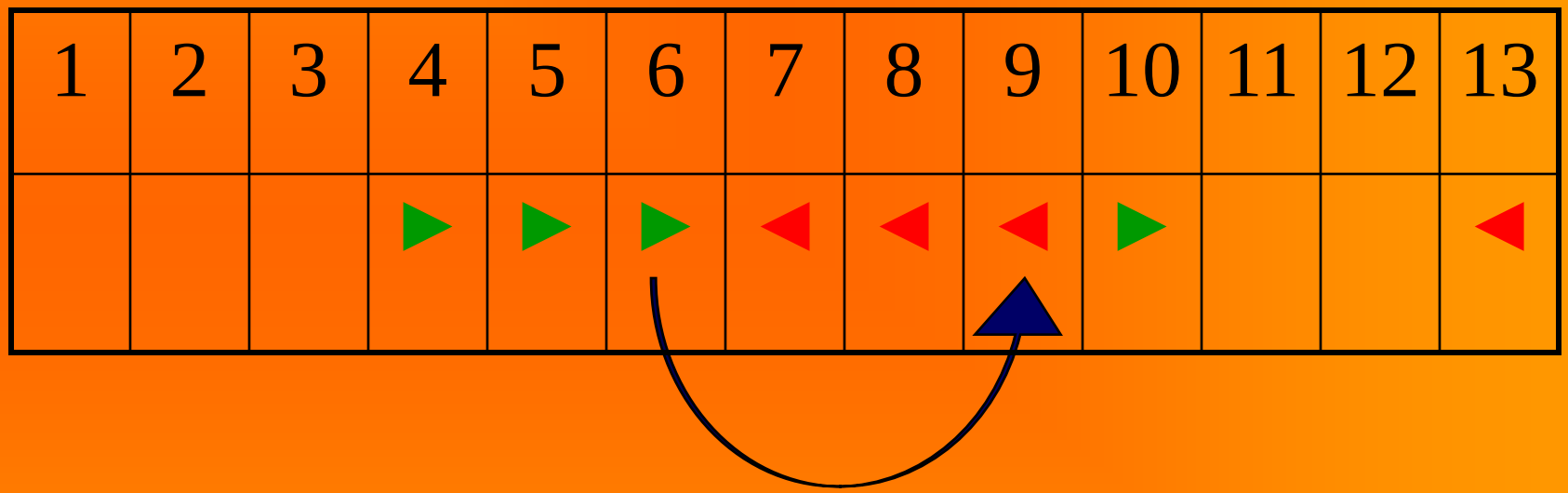
Parallelization of method I (distance = 3 , phase 2)



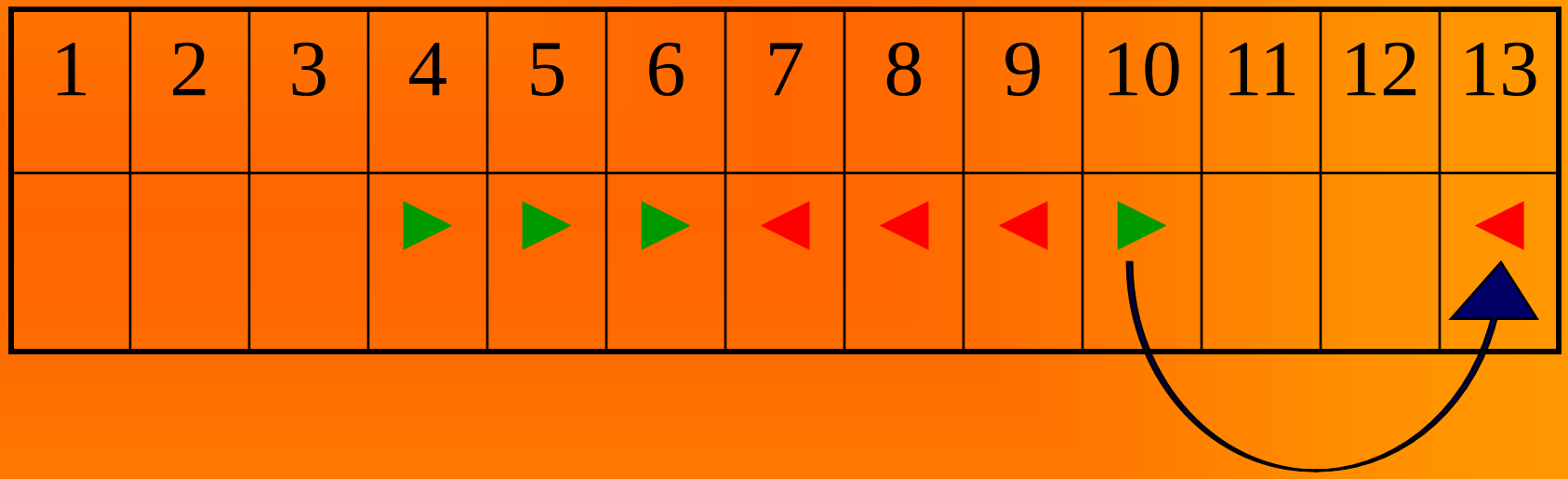
Parallelization of method I (distance = 3 , phase 2)



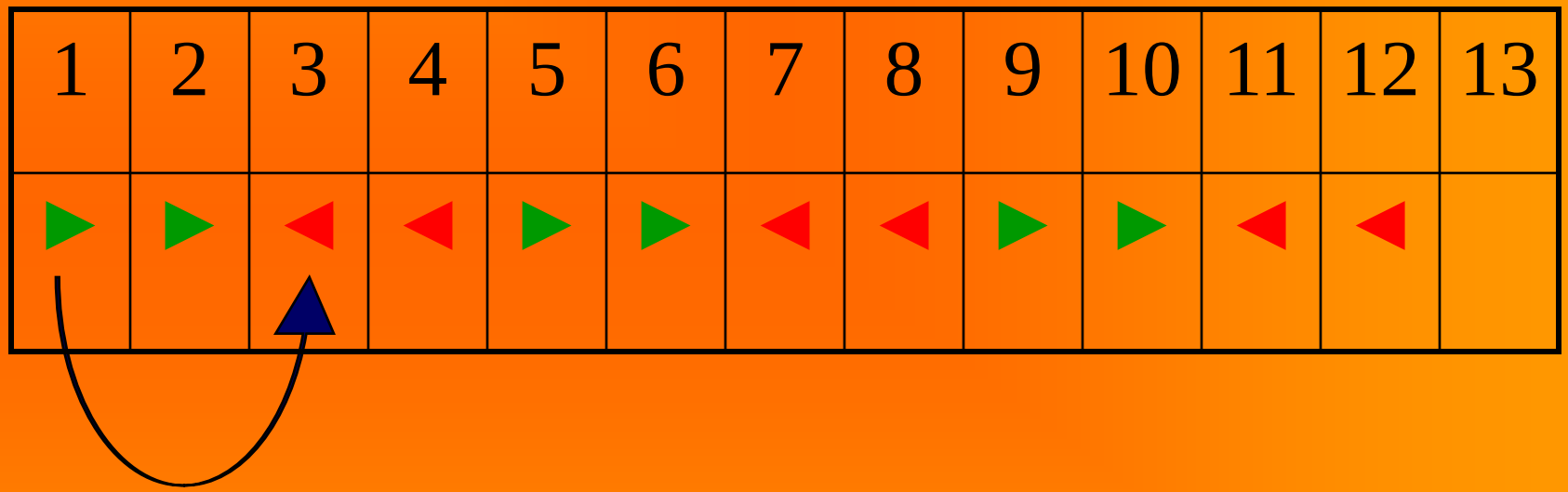
Parallelization of method I (distance = 3 , phase 2)



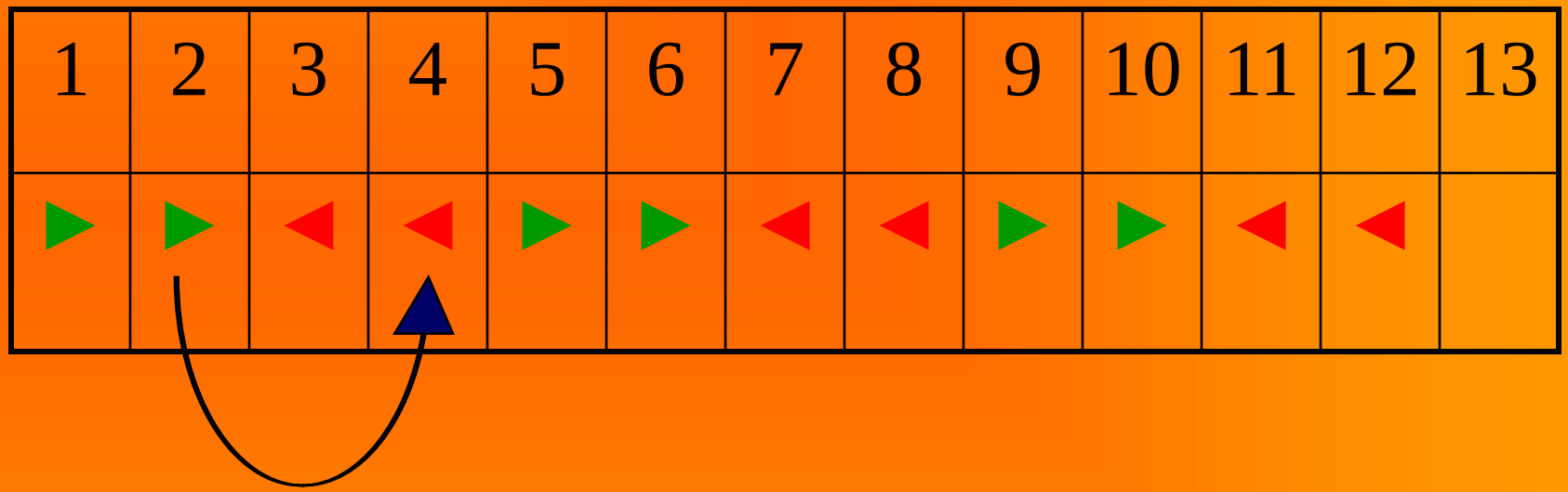
Parallelization of method I (distance = 3 , phase 2)



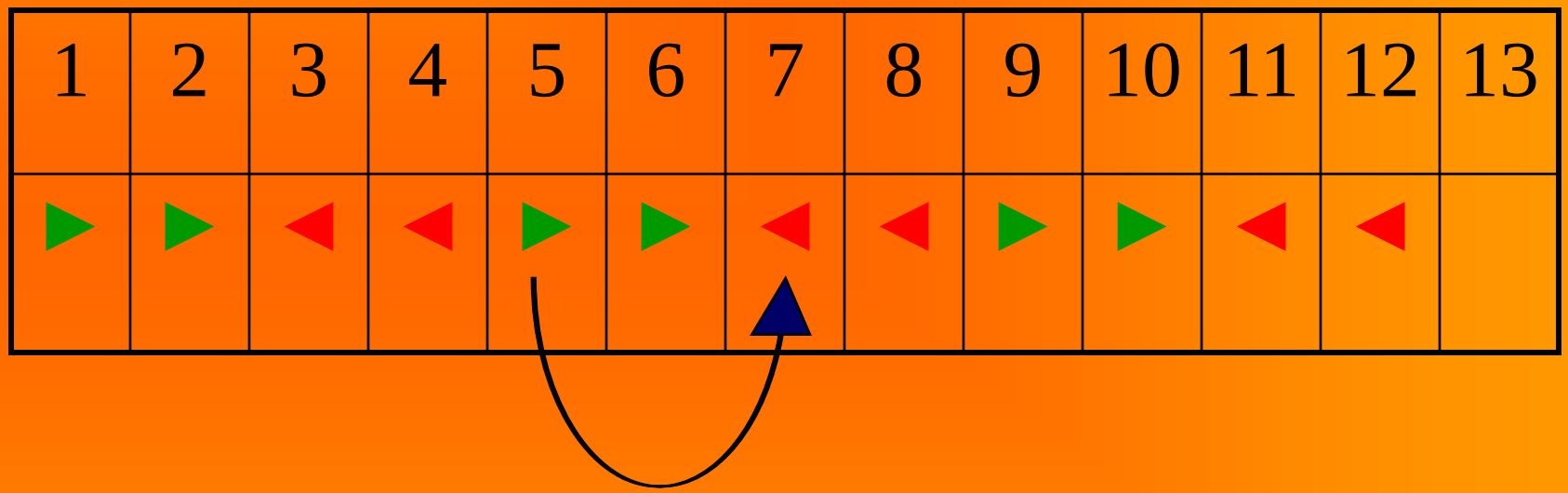
Parallelization of method I (distance = 2 , phase 1)



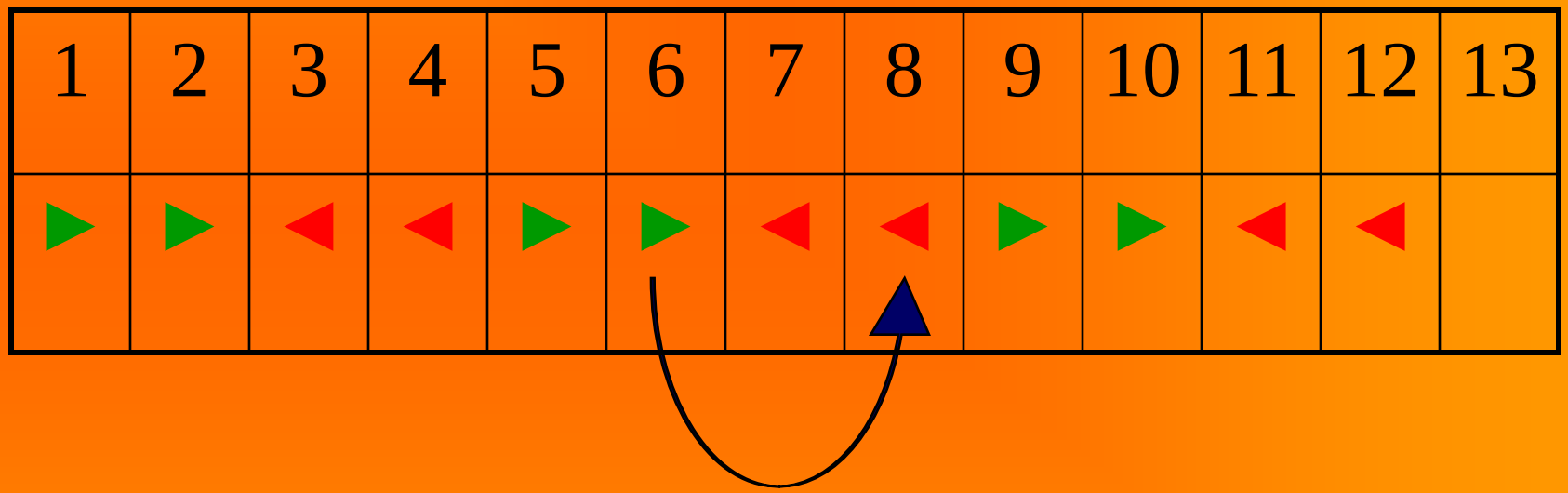
Parallelization of method I (distance = 2 , phase 1)



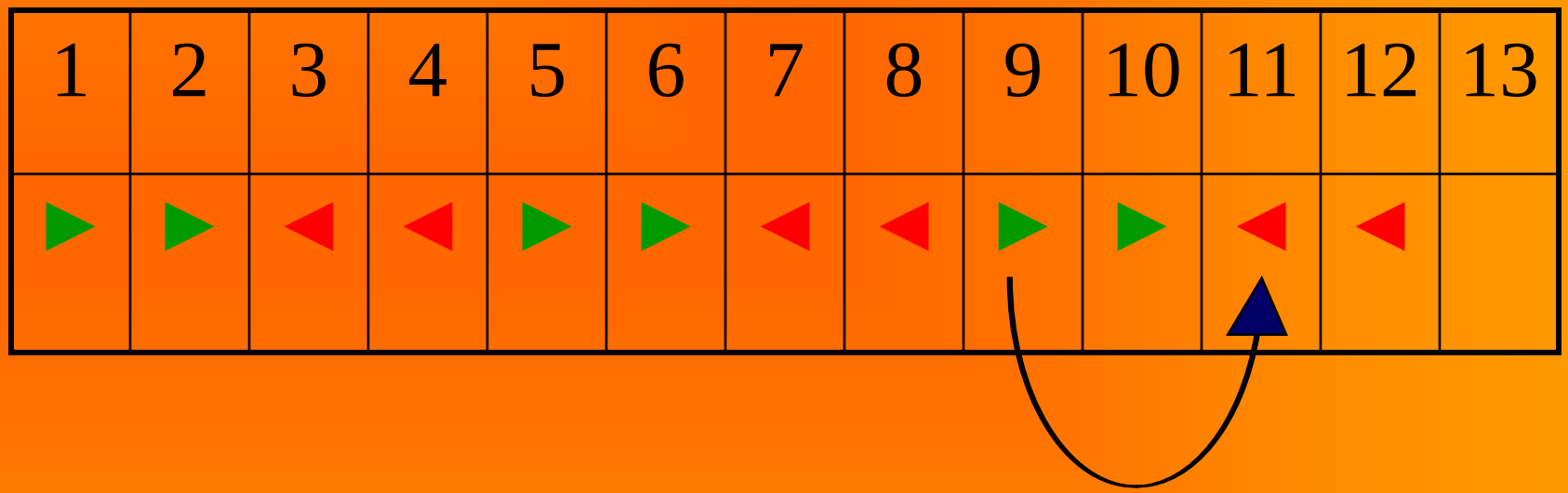
Parallelization of method I (distance = 2 , phase 1)



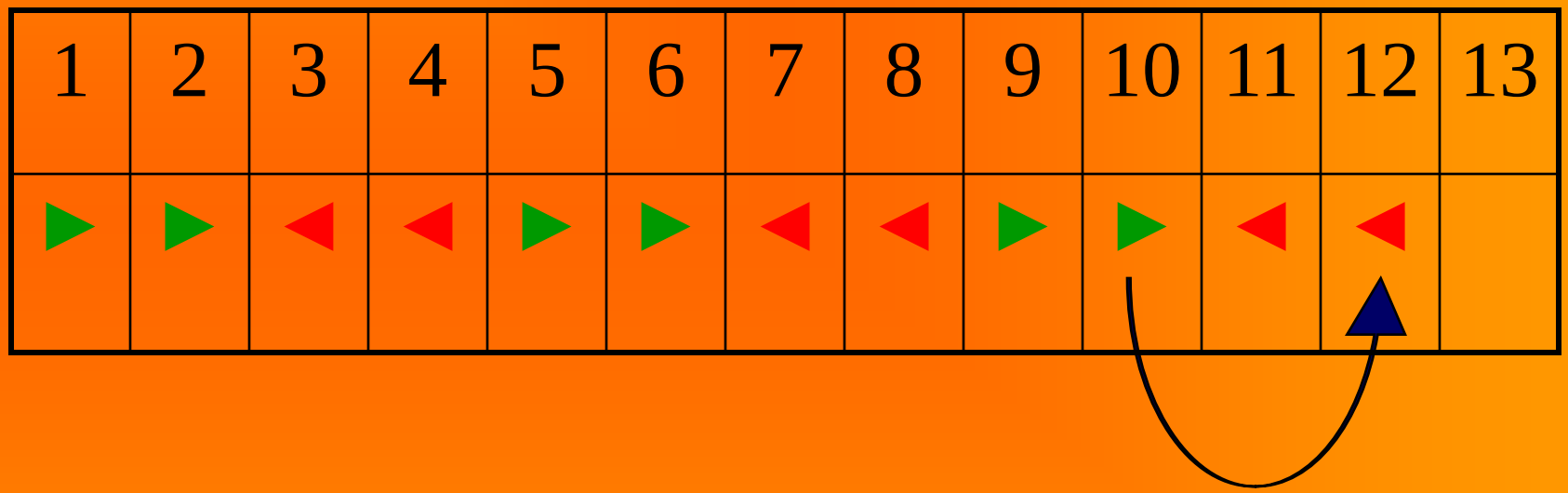
Parallelization of method I (distance = 2 , phase 1)



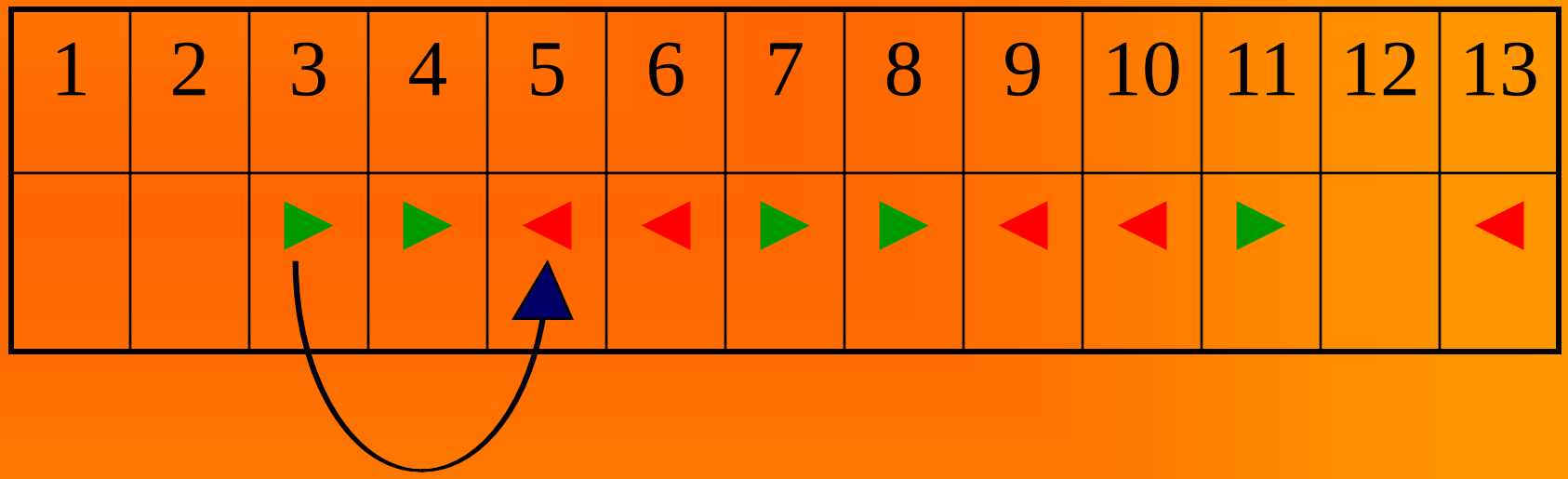
Parallelization of method I (distance = 2 , phase 1)



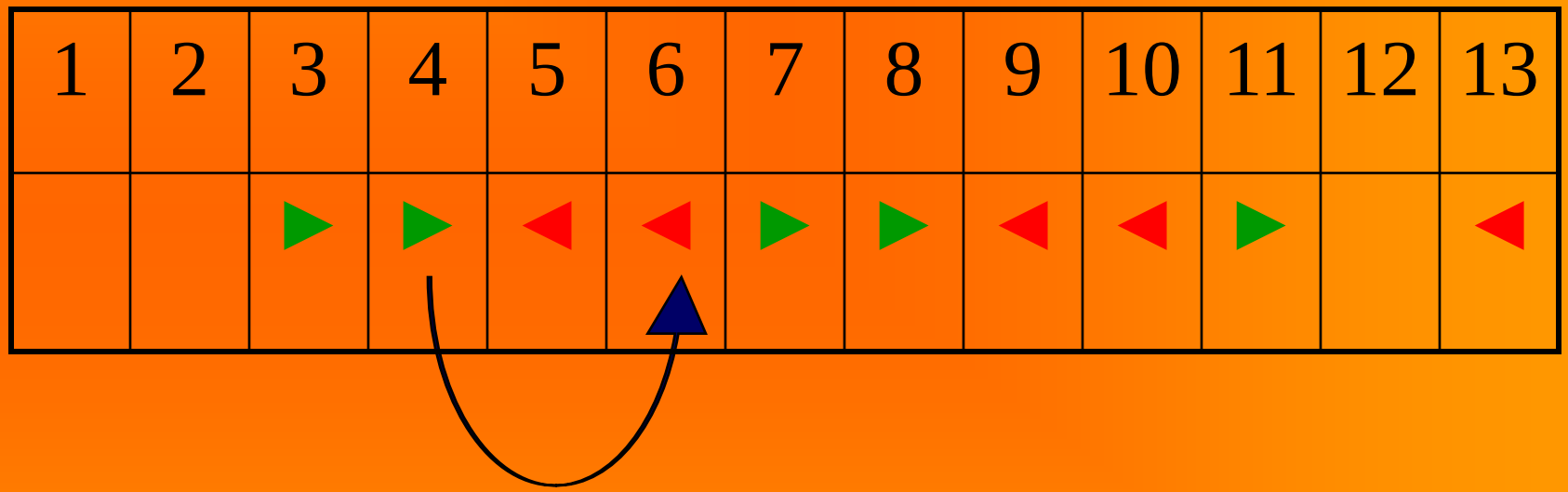
Parallelization of method I (distance = 2 , phase 1)



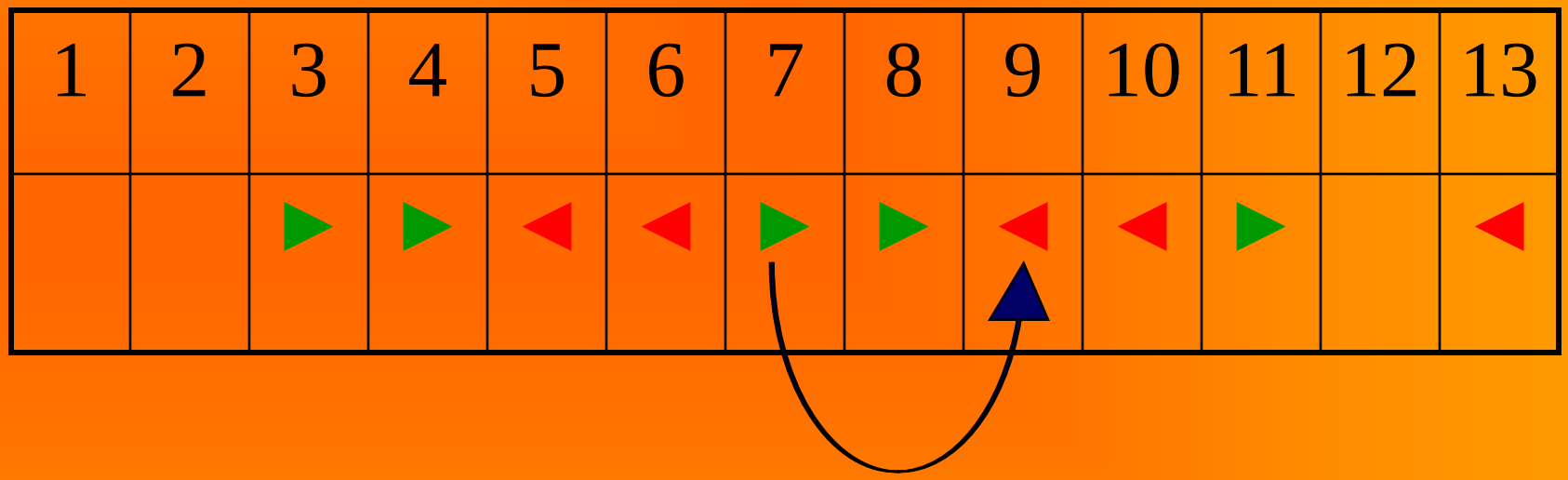
Parallelization of method I (distance = 2 , phase 2)



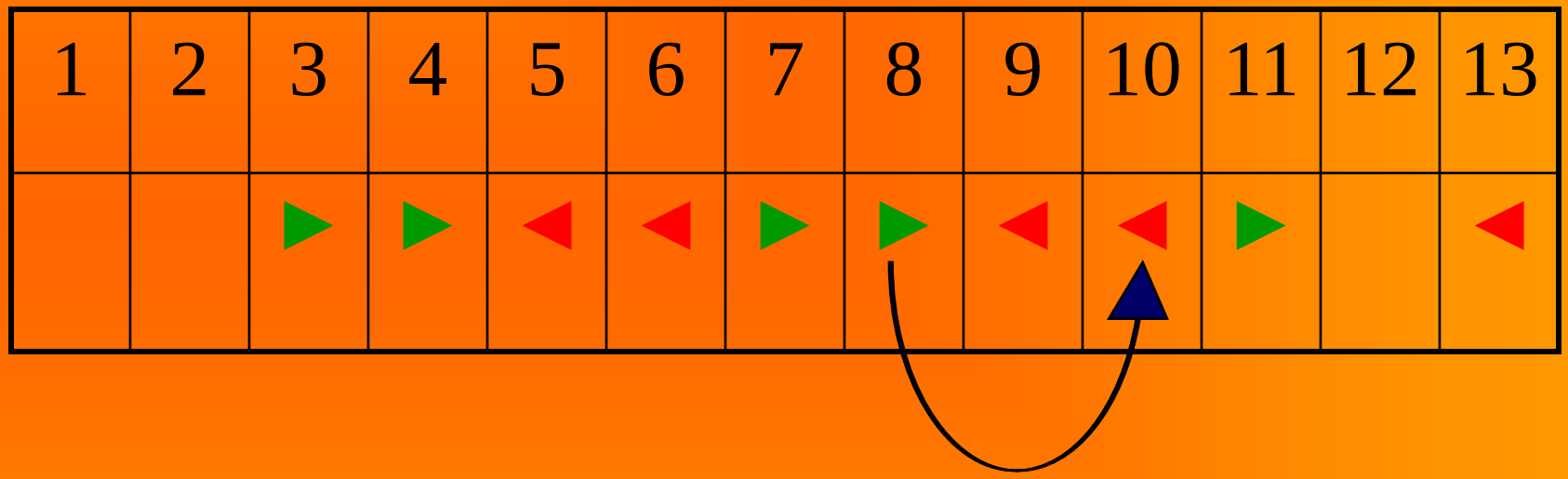
Parallelization of method I (distance = 2 , phase 2)



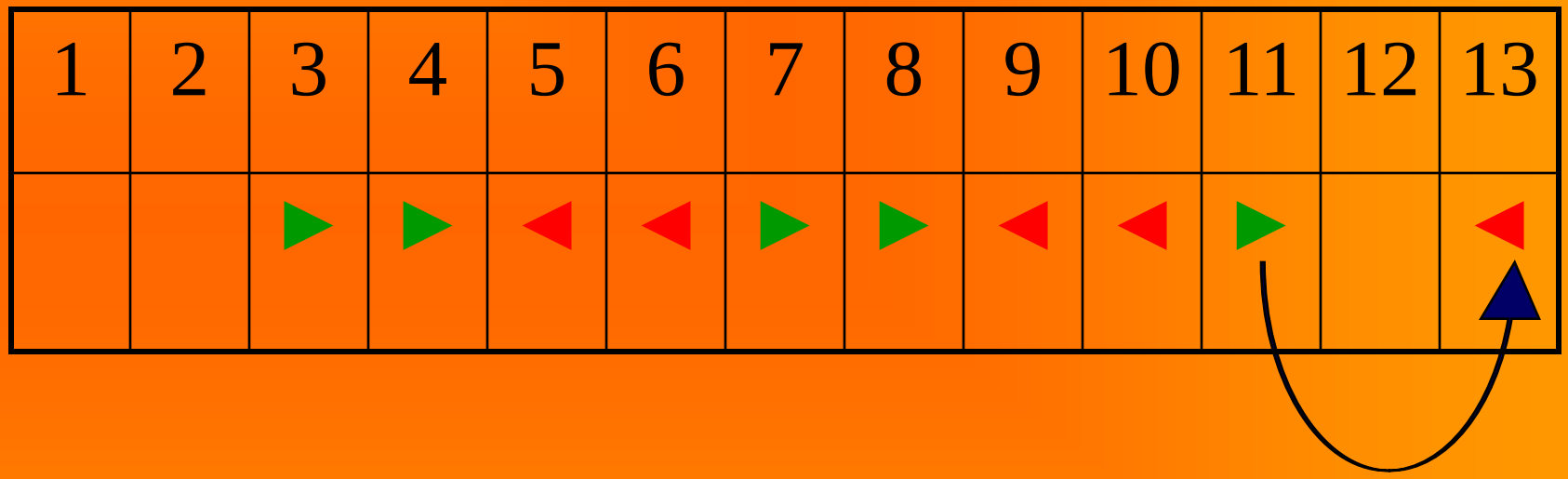
Parallelization of method I (distance = 2 , phase 2)



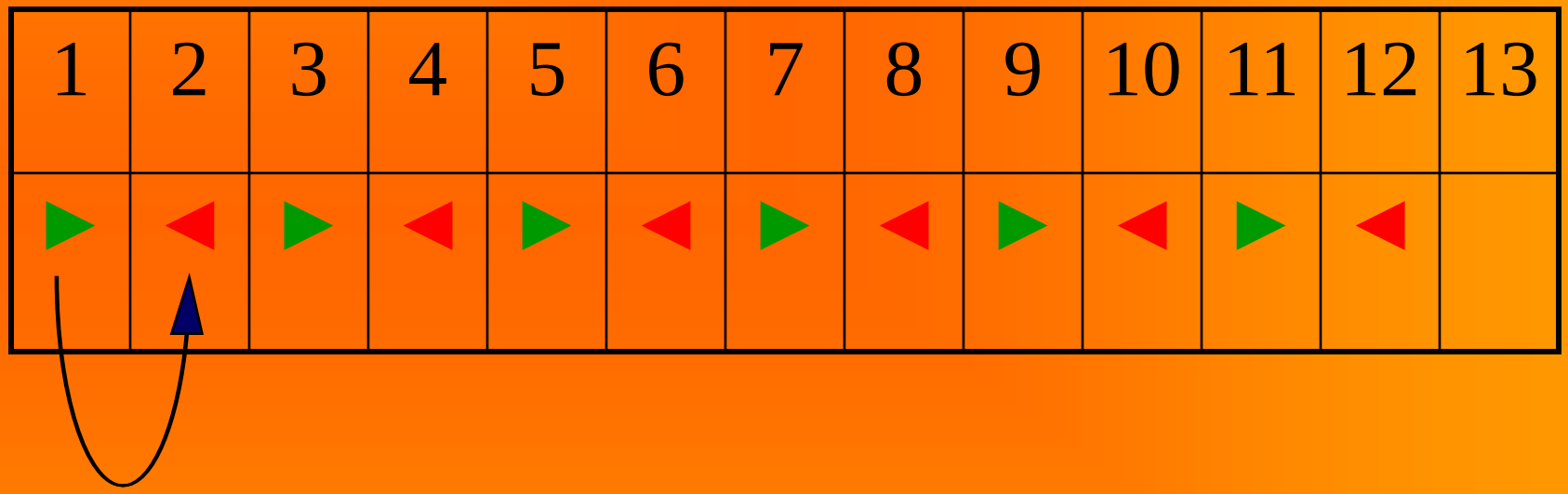
Parallelization of method I (distance = 2 , phase 2)



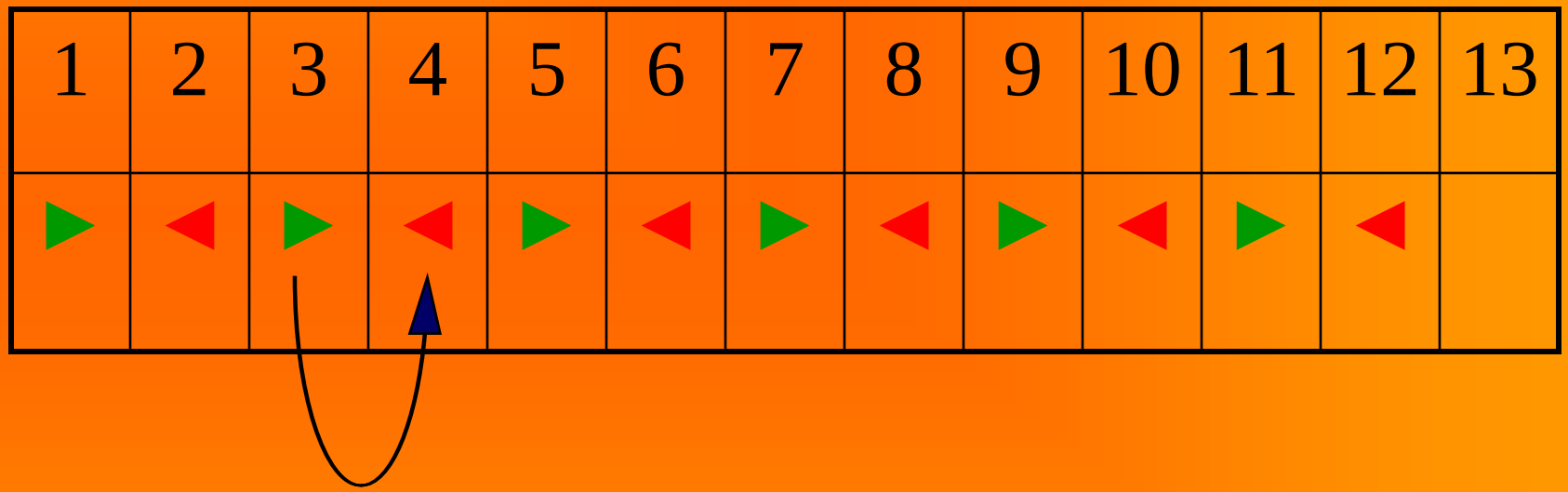
Parallelization of method I (distance = 2 , phase 2)



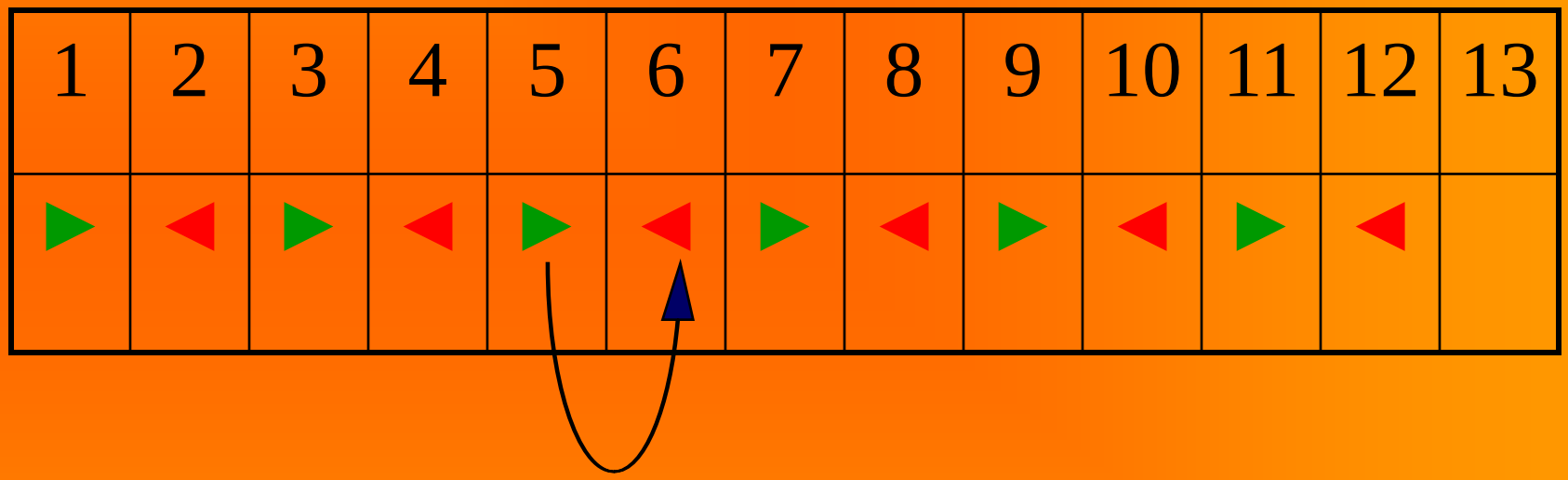
Parallelization of method I (distance = 1 , phase 1)



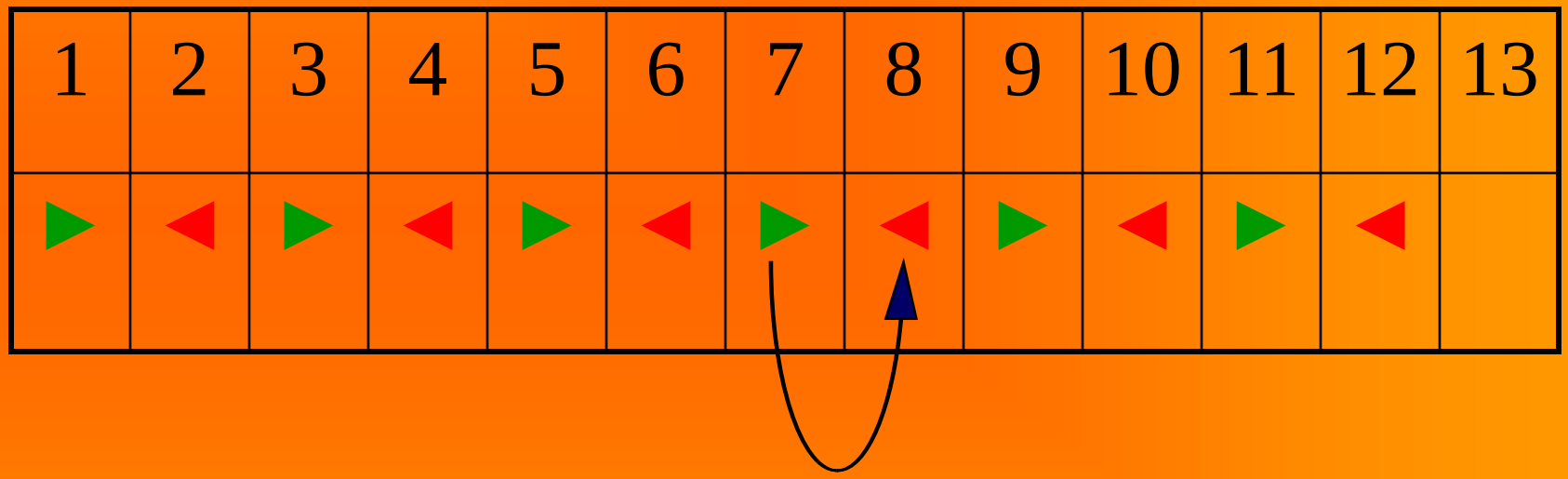
Parallelization of method I (distance = 1 , phase 1)



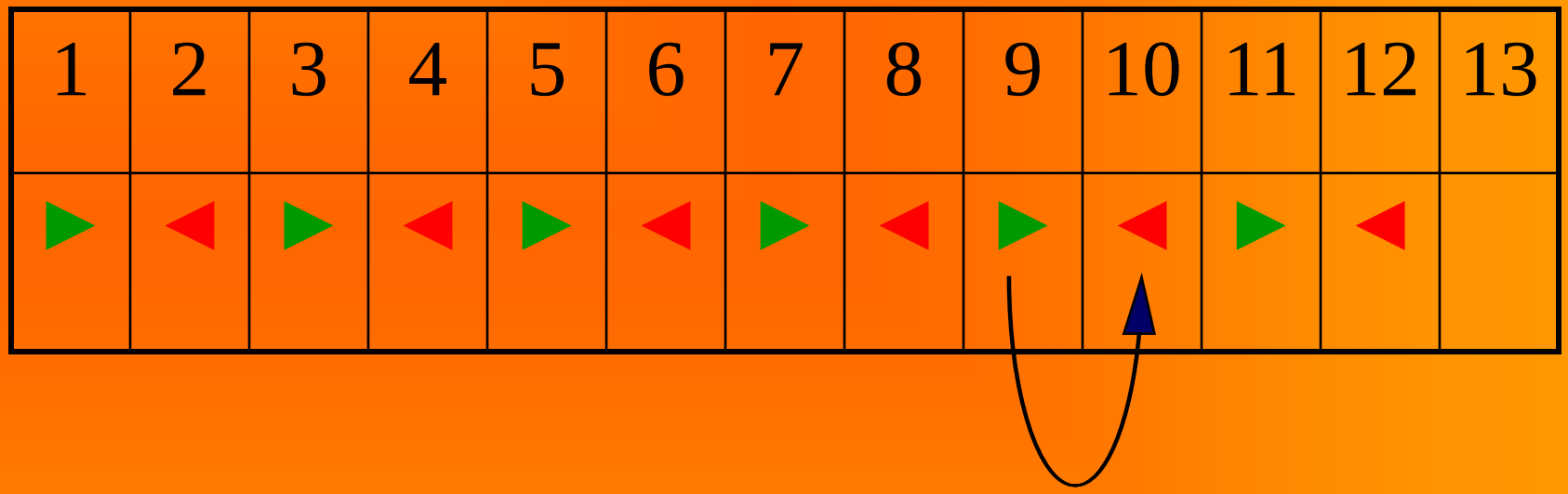
Parallelization of method I (distance = 1 , phase 1)



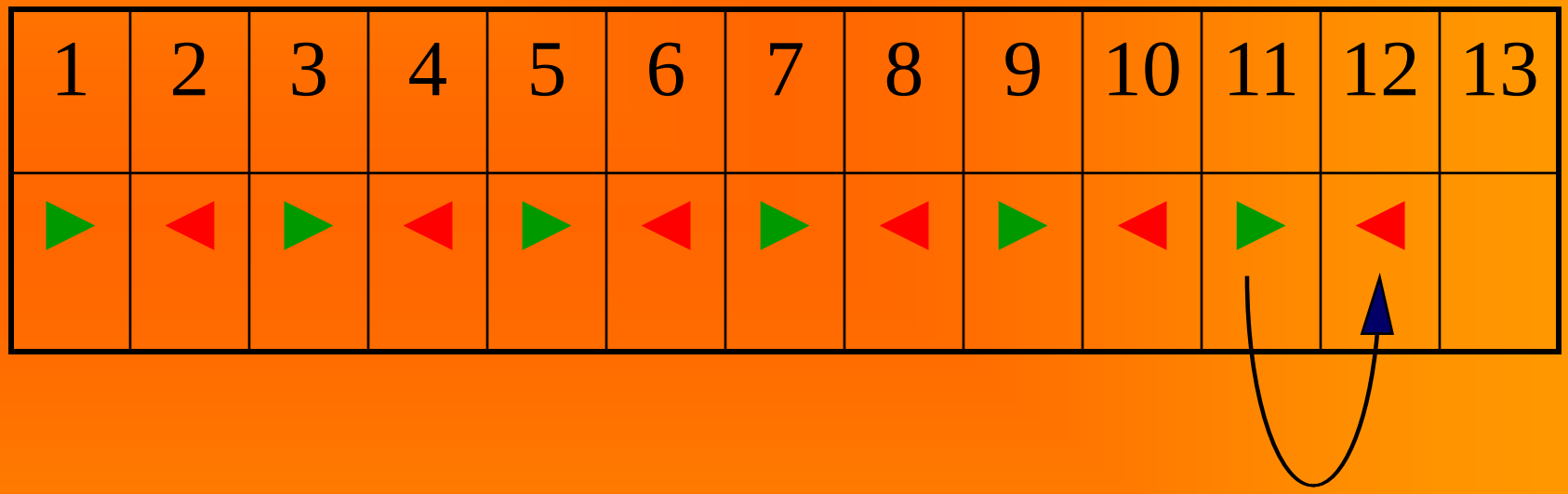
Parallelization of method I (distance = 1 , phase 1)



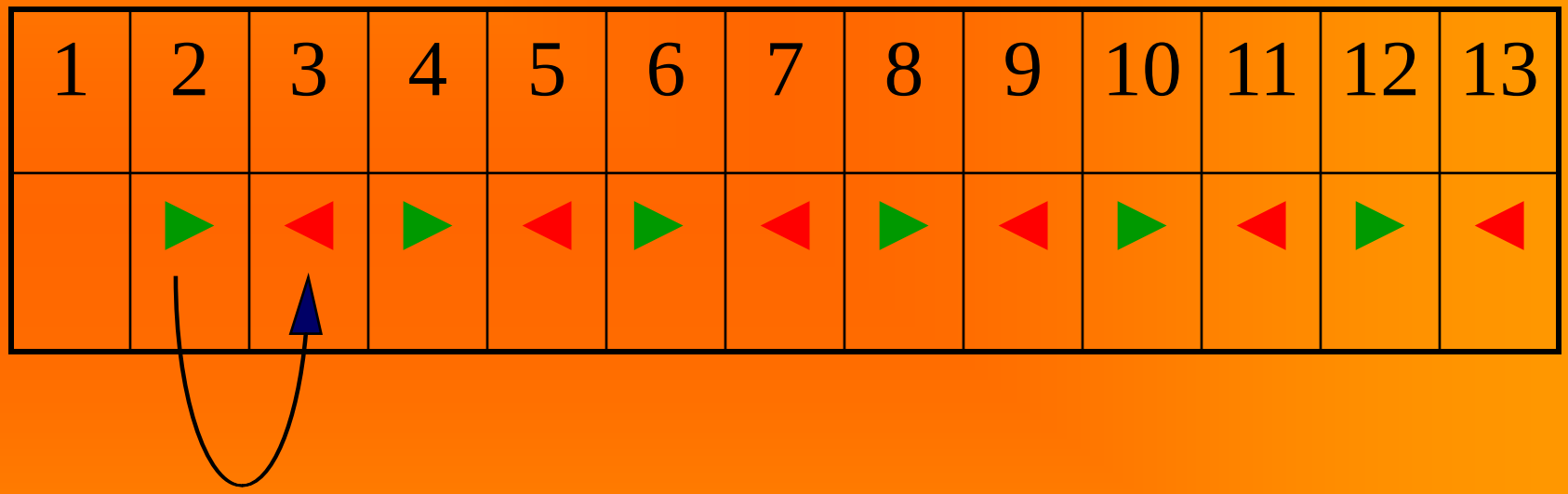
Parallelization of method I (distance = 1 , phase 1)



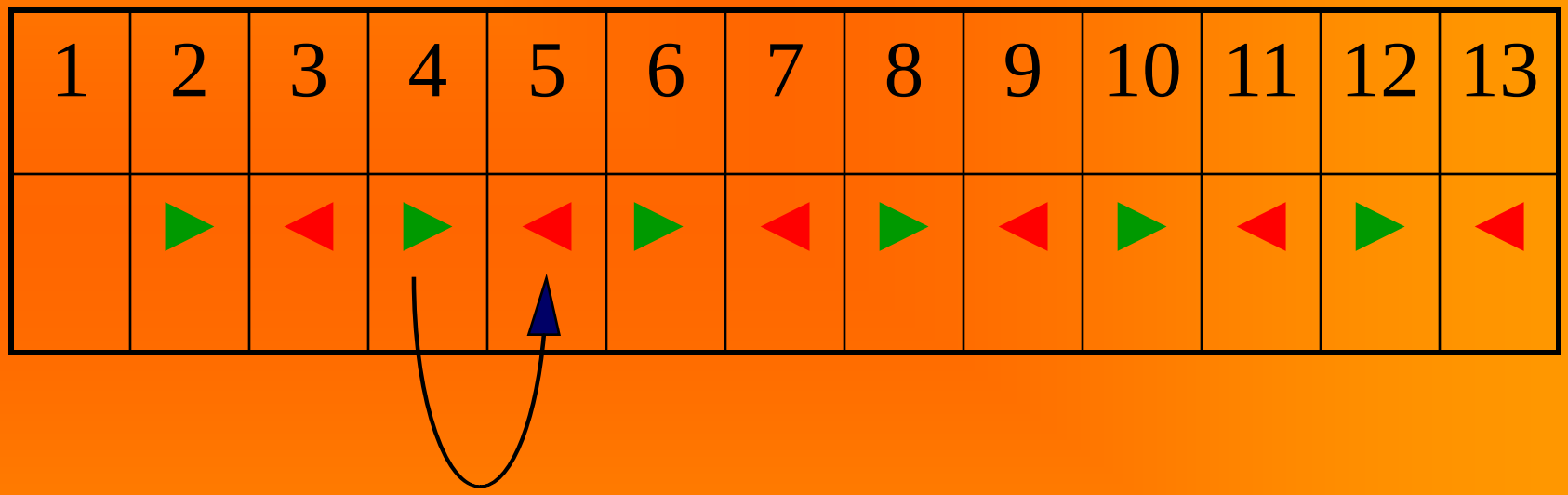
Parallelization of method I (distance = 1 , phase 1)



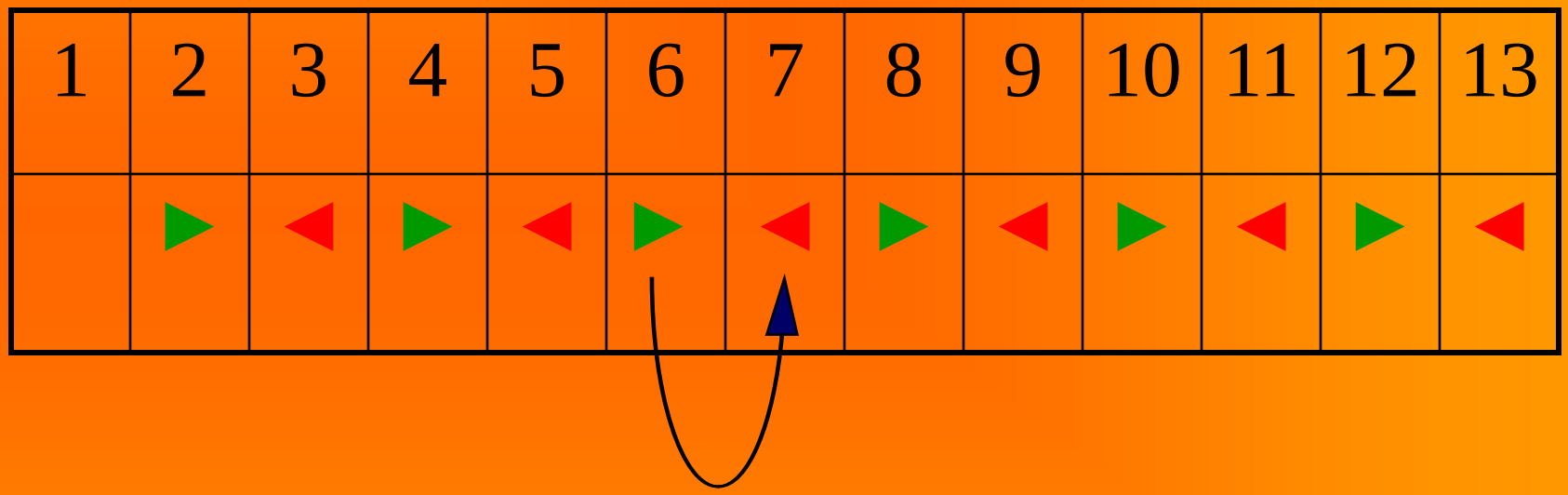
Parallelization of method I (distance = 1 , phase 2)



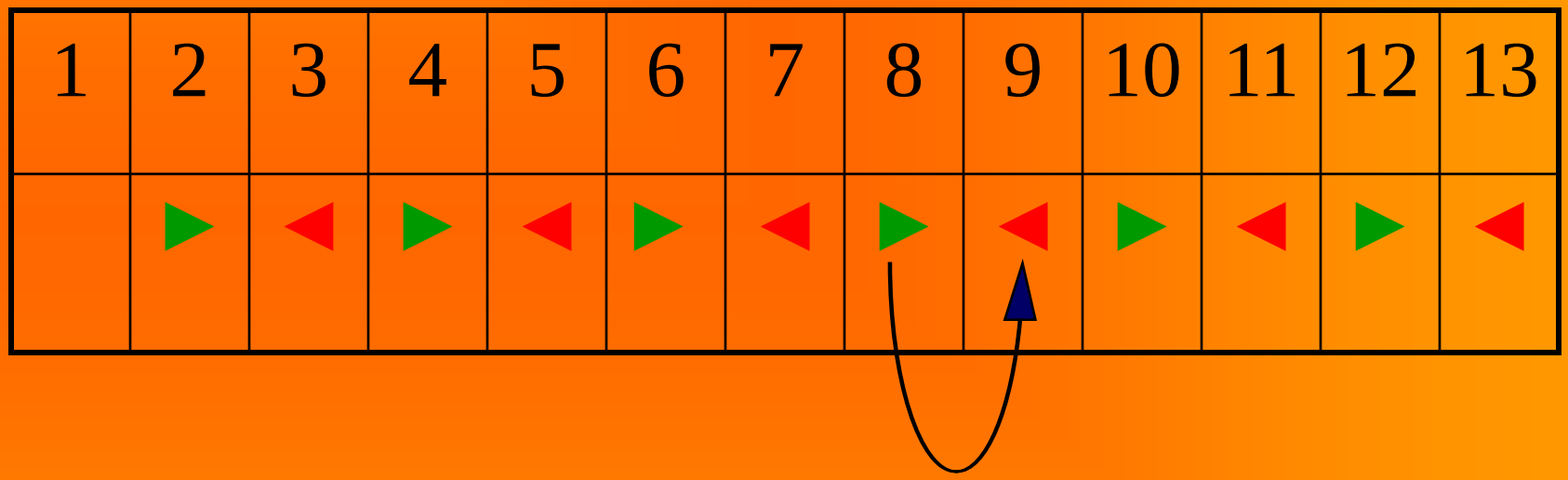
Parallelization of method I (distance = 1 , phase 2)



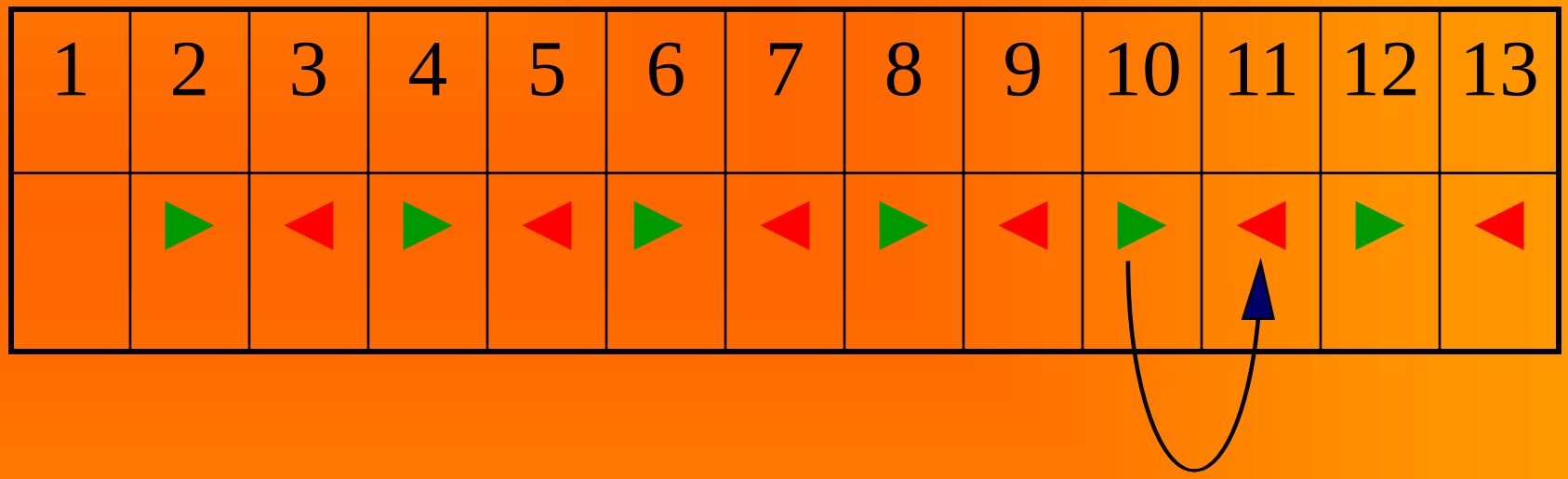
Parallelization of method I (distance = 1 , phase 2)



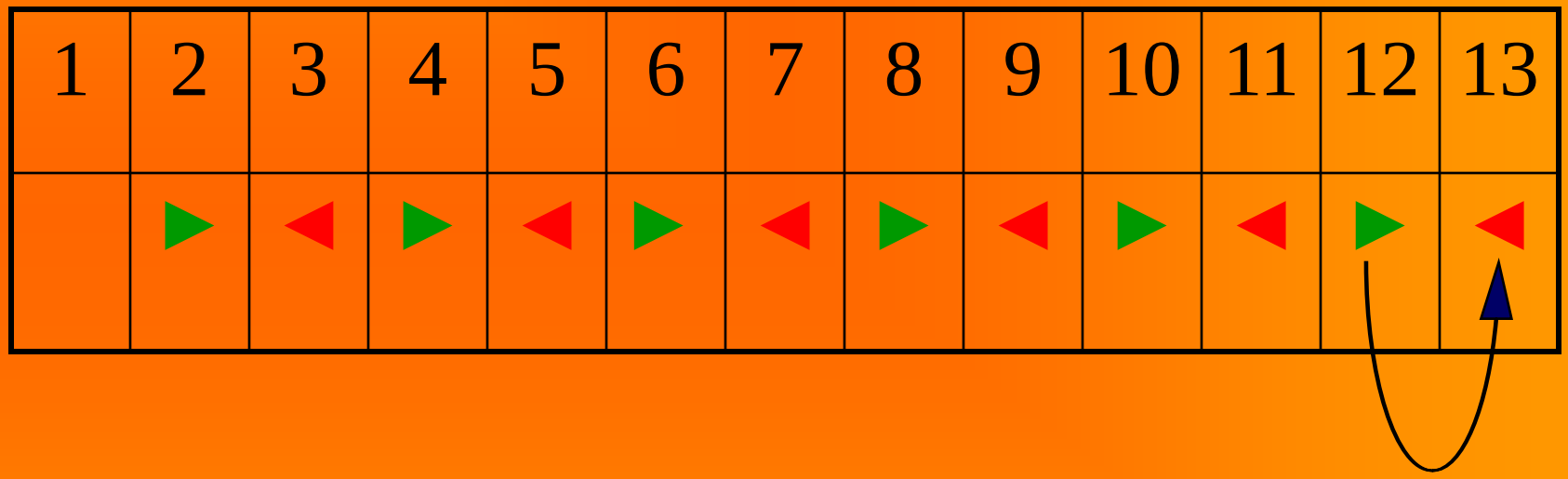
Parallelization of method I (distance = 1 , phase 2)



Parallelization of method I (distance = 1 , phase 2)



Parallelization of method I (distance = 1 , phase 2)



Parallelization of method I (summary)

Repeat above steps
(distance: 12, 11, 10, 9, 8, 7:
no phases
distance: 6, 5, 4, 3, 2, 1:
phase 1 and phase 2)
until no swapping is done

Computers

„BARIBAL” - SGI Altix 3700

Operating system:

SUSE Linux Enterprise Server 10

Processors: 256 x Intel Itanium 2

1.5 GHz clock, 512 GB memory

„PANDA” - SGI Altix 4700

Operating system:

SUSE Linux Enterprise Server 10

Processors: 32 x Intel Itanium 2

1.66 GHz clock, 64 GB memory

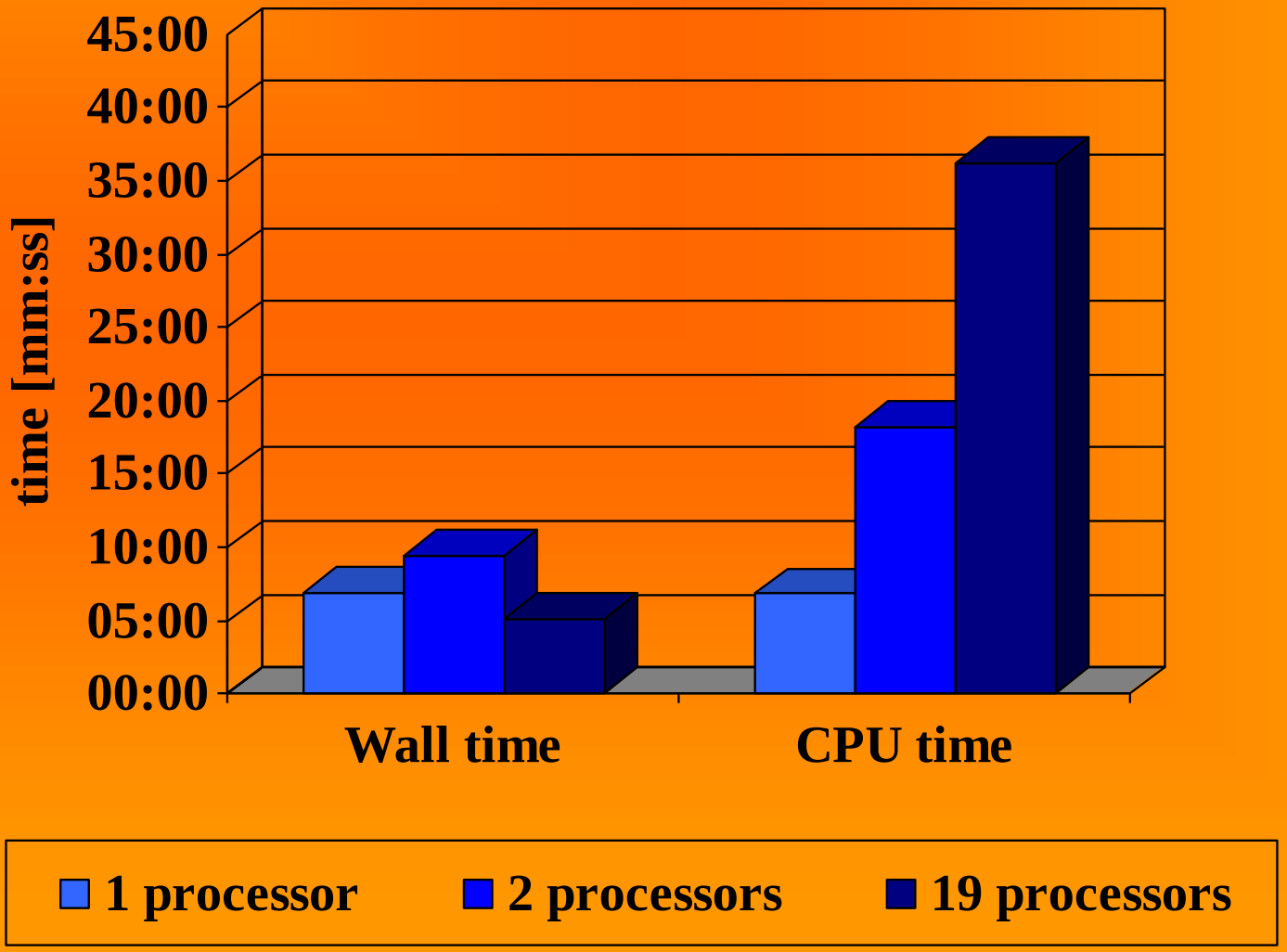
Software

- Intel Fortran
- !$OMP PARALLEL DO directive
- DYNAMIC scheduling
- „CHUNK size” 1000

Results - „*Baribal*”

Number of processors	Wall time	CPU time
1	06'51''	06'48''
2	09'28''	18'11''
19	05'03''	36'08''

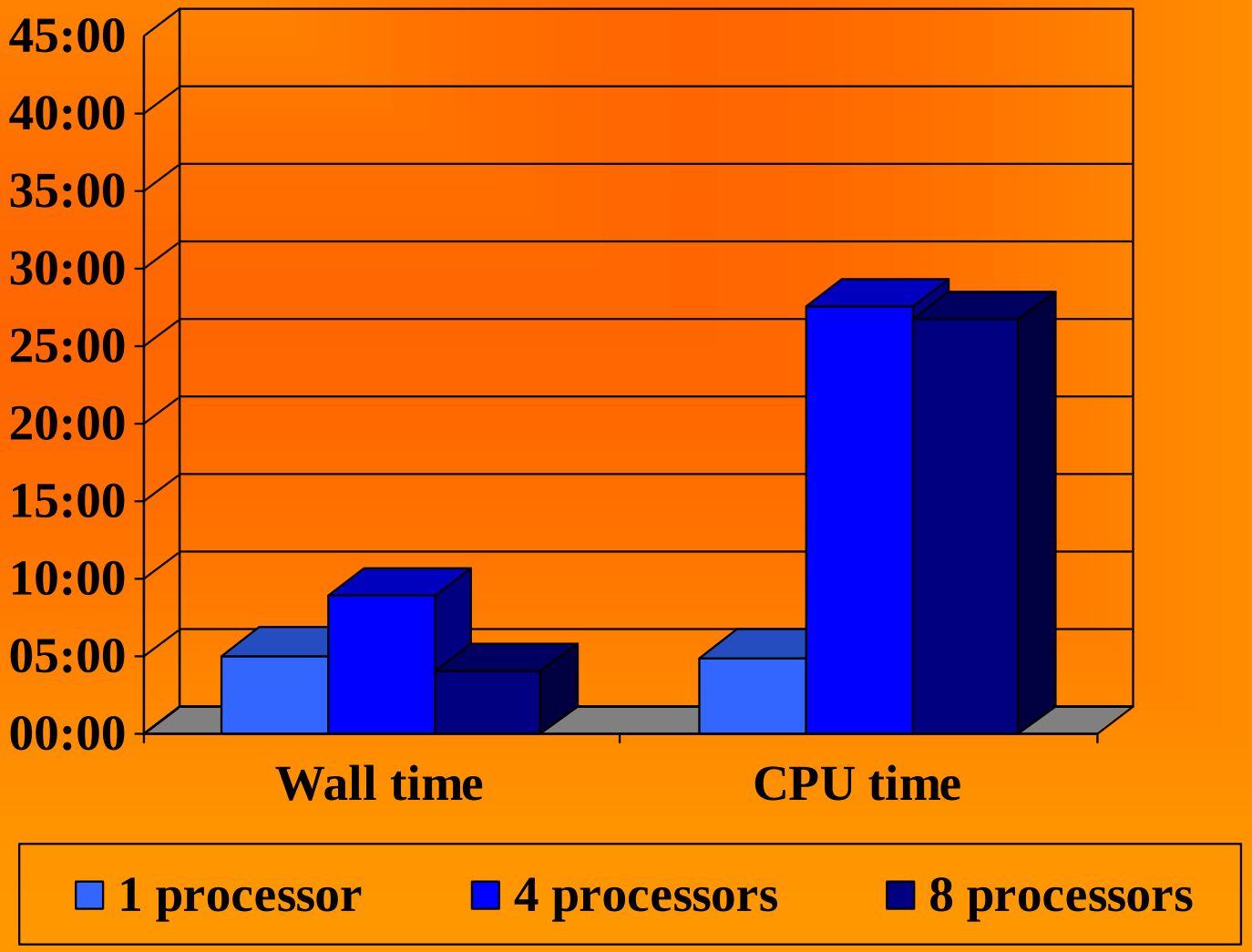
Results - „*Baribal*” (II)



Results - „*Panda*”

Number of processors	Wall time	CPU time
1	05'03"	04'55"
4	08'51"	27'32"
8	03'59"	26'43"

Results - „*Panda*” (II)



Conclusions

- *Parallelization of the algorithm let decrease execution time.*
- *The speedup factor, however, was not satisfying; assuming real execution time for 1-processor version to be 100%, the corresponding times for multiprocessor versions decreased only to about 75% - 80%, depending on the number of processors.*
- *The algorithm must be further tested with larger number of processors and its efficiency must be improved.*

Appendix (1) ☺

DKR-86022	PKR.O-CXCIX-48426	PKR.0-CXLI-36036	suka
KW.I-142/OP			suka
KW.I-176/OP	KW.T-IV-122/28	PKR.I-48874	pies
KW.I-179/OP			pies
KW.I-188/OP			pies
KW.I-189/OP			suka
KW.I-191/OP			suka
KW.I-218/OP			suka
KW.I-229/OP	KW.I-188/OP	KW.I-189/OP	suka
KW.I-232/OP	PKR.I-38367	KW.I-142/OP	pies
KW.I-233/OP			pies
KW.I-239/OP	KW.I-188/OP	KW.I-189/OP	suka
KW.I-240/OP	KW.I-188/OP	KW.I-189/OP	pies
KW.I-248/OP			pies
KW.I-249/OP	PKR.I-37301	KW.I-218/OP	suka

Appendix (2) ☺

PKR.O-CXLI-36036			suka
KW.I-142/OP			suka
KW.T-IV-122/28			pies
KW.I-179/OP			pies
KW.I-188/OP			pies
KW.I-189/OP			suka
KW.I-191/OP			suka
KW.I-218/OP			suka
KW.I-229/OP	KW.I-188/OP	KW.I-189/OP	suka
KW.T-I-306/XXVIII			pies
KW.I-233/OP			pies
KW.I-239/OP	KW.I-188/OP	KW.I-189/OP	suka
KW.I-240/OP	KW.I-188/OP	KW.I-189/OP	pies
KW.I-248/OP			pies
KW.T-III-107/XXVIII			suka
KW.T-34/XXVIII			suka
KW.T-II-132/XXVIII			suka
KW.T-II-119/XXVIII			suka
KW.T-II-122/XXVIII			pies
KW.T-III-223/XXVIII	KW.T-I-306/XXVIII	KW.T-II-132/XXVIII	pies
PKR.O-LXXII- 19853			pies
KW.T-III-119/XXVIII			suka

Appendix (3) ☺

```
..-176[...]Lupo Baca Chluba Liliowej Przełęczy
  KW.[....] Baca [...]
  PKR.[...] Kobza Chluba Liliowej Przełęczy
```

Thank you for your attention.