Genetic Algorithms

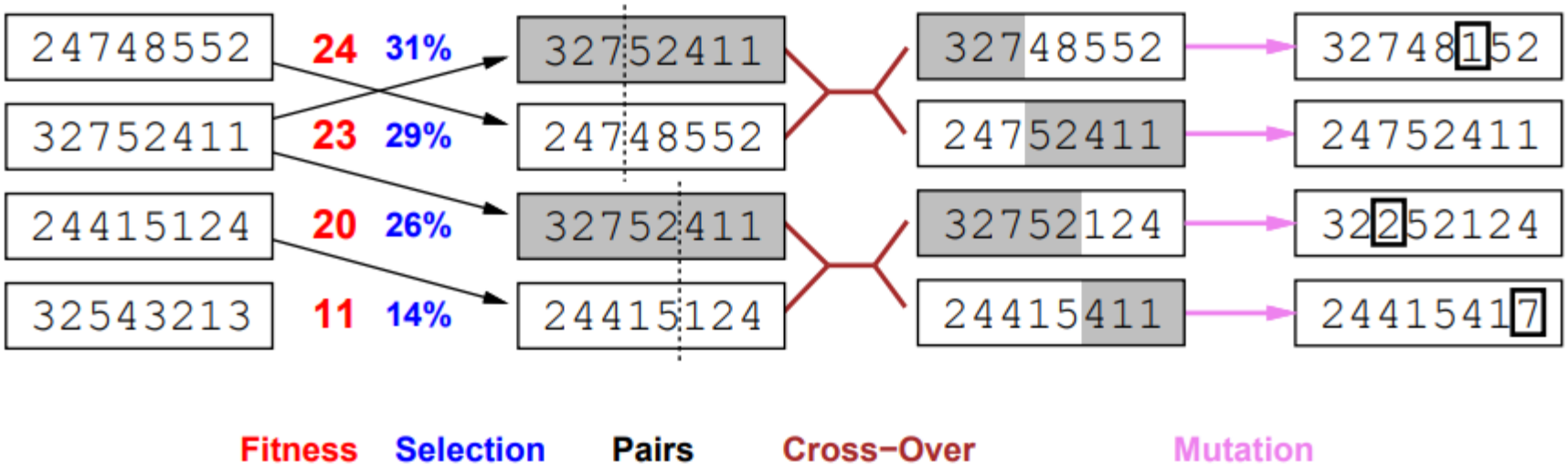
Chapter 4b

(based on AIMA Slides)

Genetic Algorithms

= stochastic local beam search

+ generate successors from pairs of states



Genetic Algorithms

- In computer science and operations research, a **genetic algorithm (GA)** is a metaheuristic inspired by the process of natural selection that belongs to the larger class of evolutionary algorithms (EA). **Genetic algorithms** are commonly used to generate high-quality solutions to optimization and search problems by relying on bio-inspired operators such as mutation, crossover and selection.

Genetic Algorithms

- A **Genetic Algorithm** is a type of local search that mimics evolution by taking a population of strings, which encode possible solutions and combines them based on a fitness function to produce individuals that are more fit.

Selection

- **Selection** is the stage of a genetic algorithm in which individual genomes are chosen from a population for later breeding (using the crossover operator).

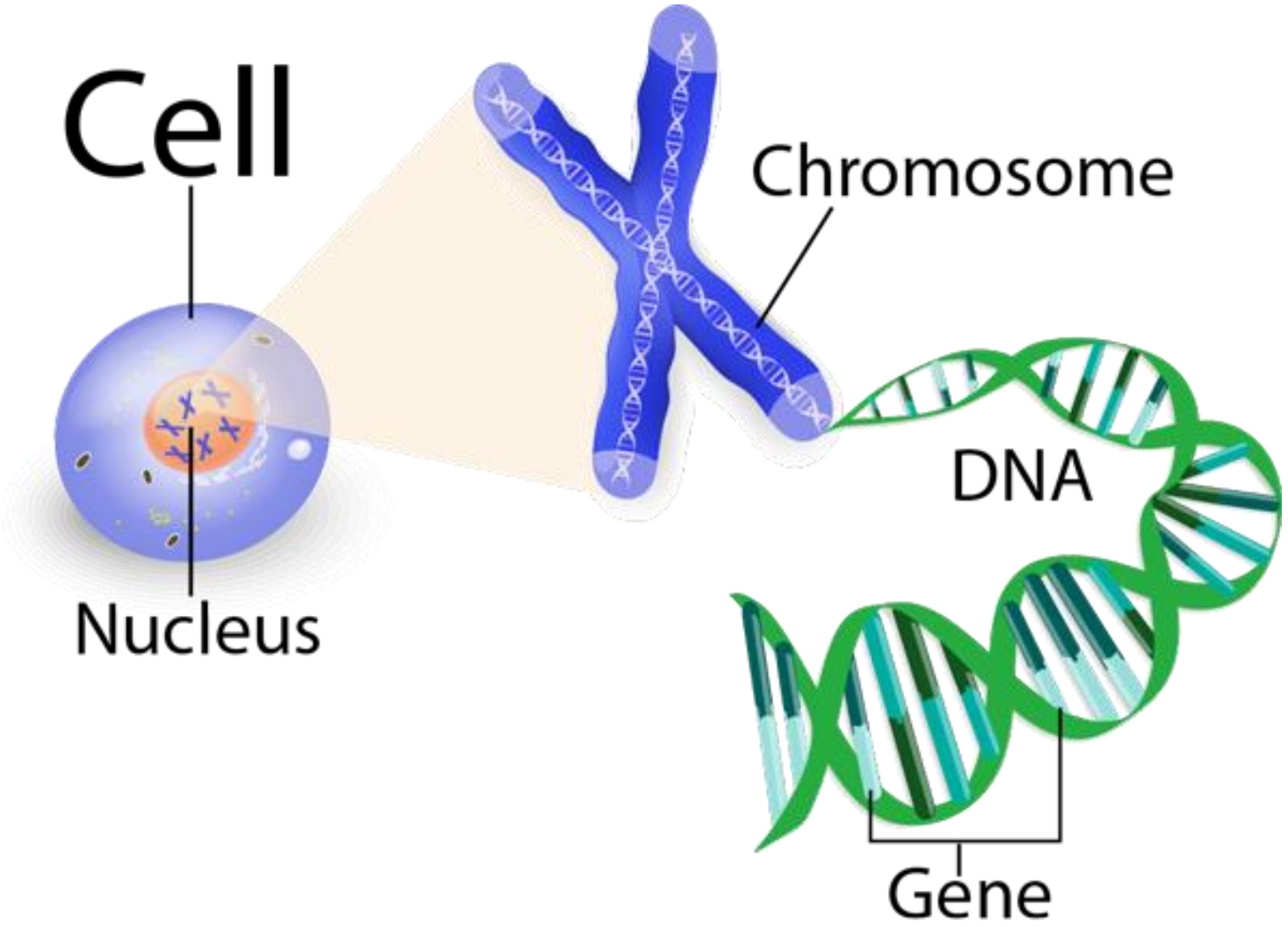
Crossover

- In GAs, **crossover** is a genetic operator used to vary the programming of a chromosome (or chromosomes) from one generation to the next. It is analogous to reproduction and biological crossover, upon which GA are based.
- **Crossover** is a process of taking more than one parent solutions and producing a child solution from them.

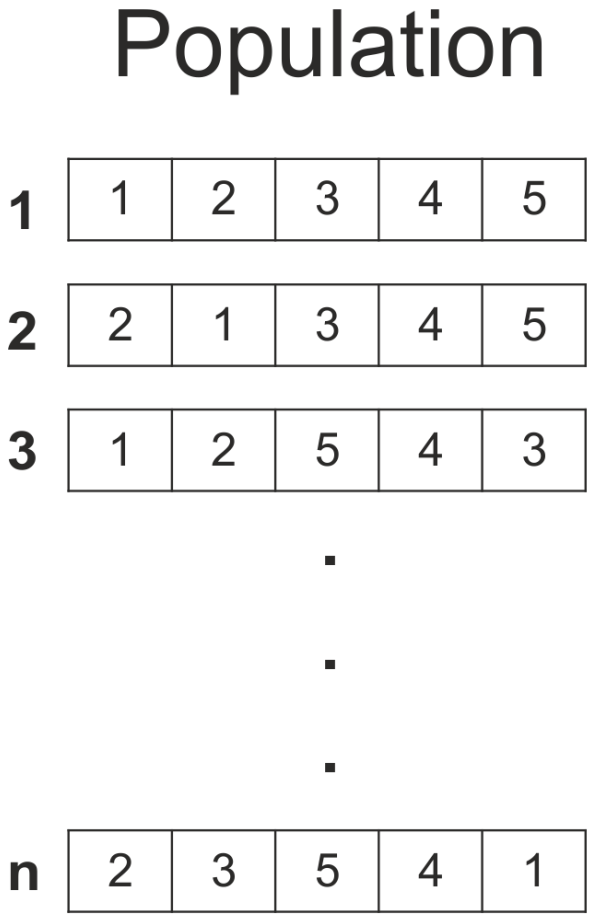
Mutation

- **Mutation** is a genetic operator used to maintain genetic diversity from one generation of a population of genetic algorithm chromosomes to the next. It is analogous to biological mutation. **Mutation** alters one or more gene values in a chromosome from its initial state.

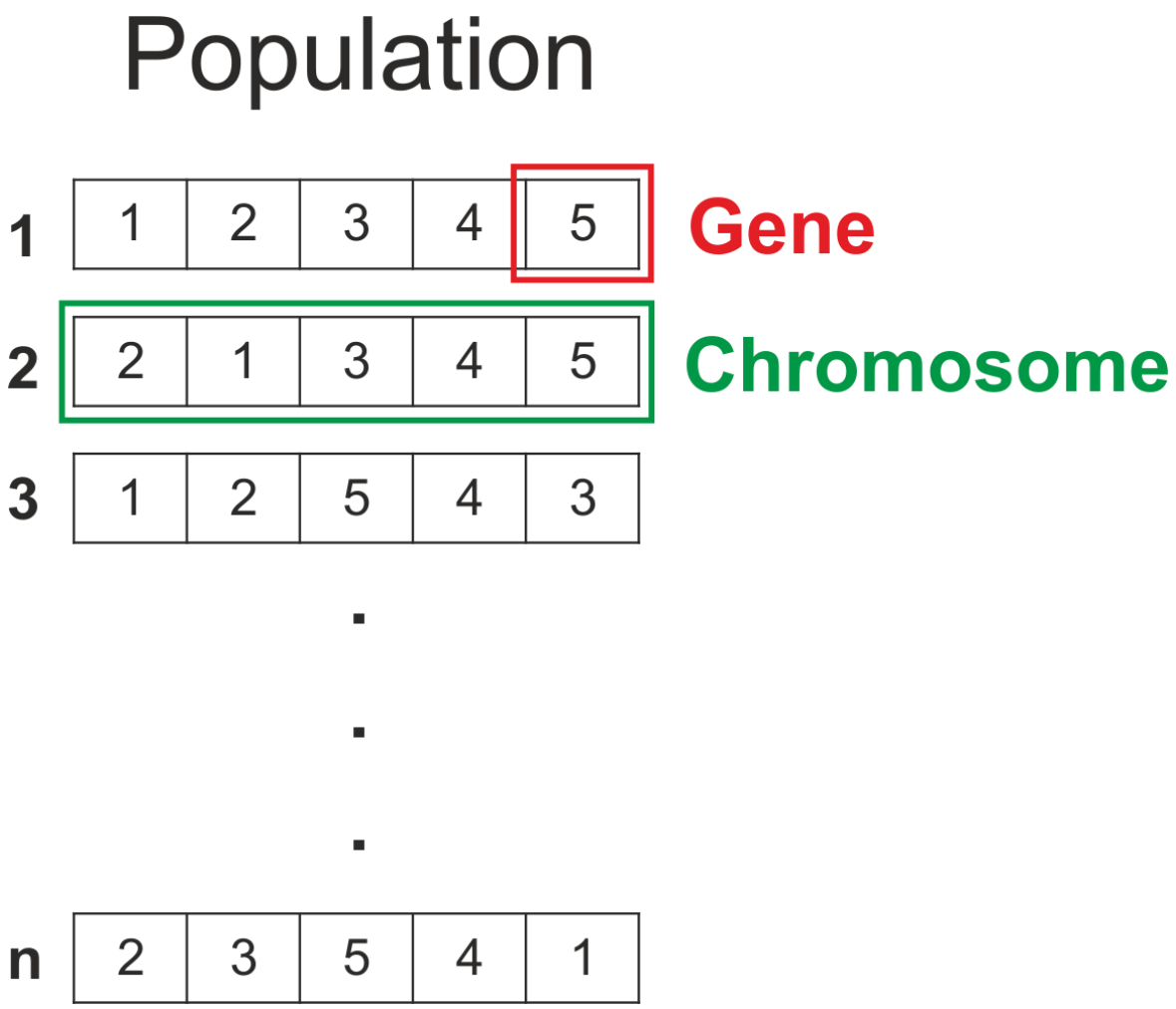
GAs & Biology



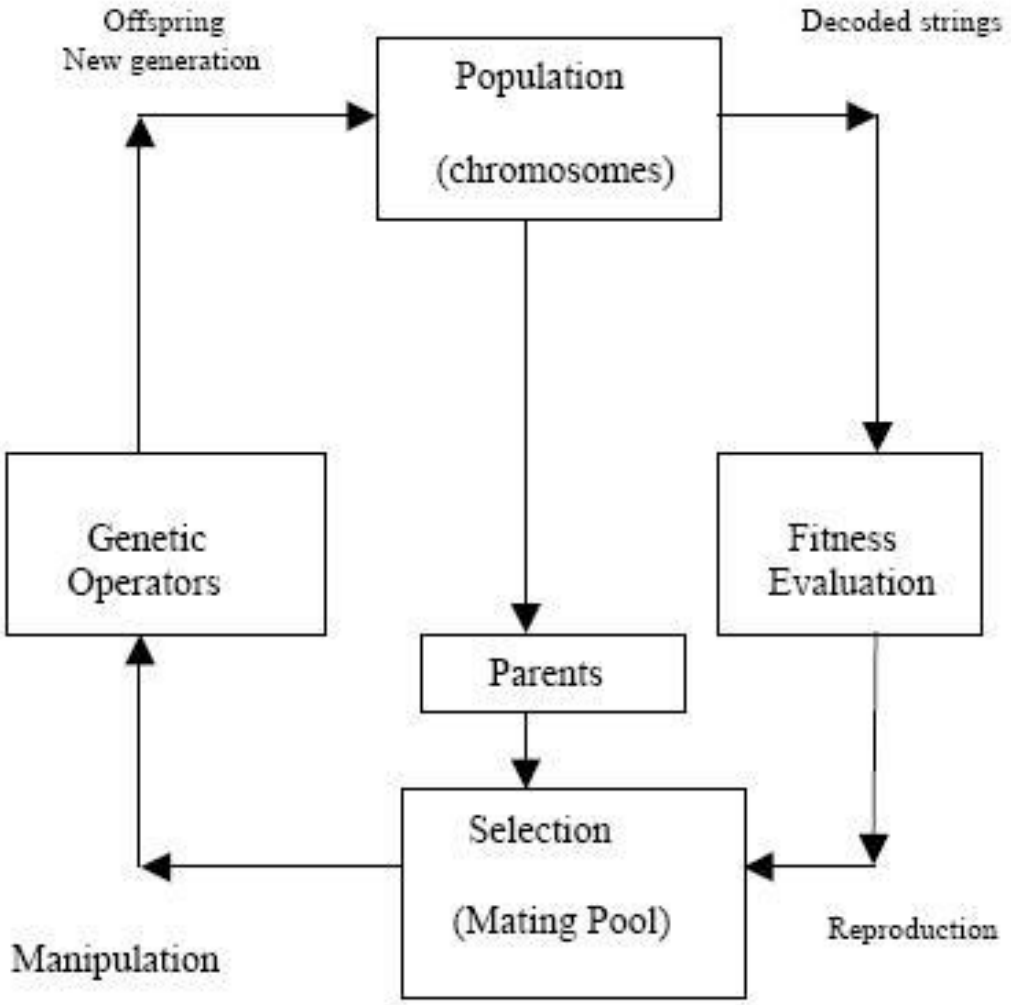
GAs & Biology



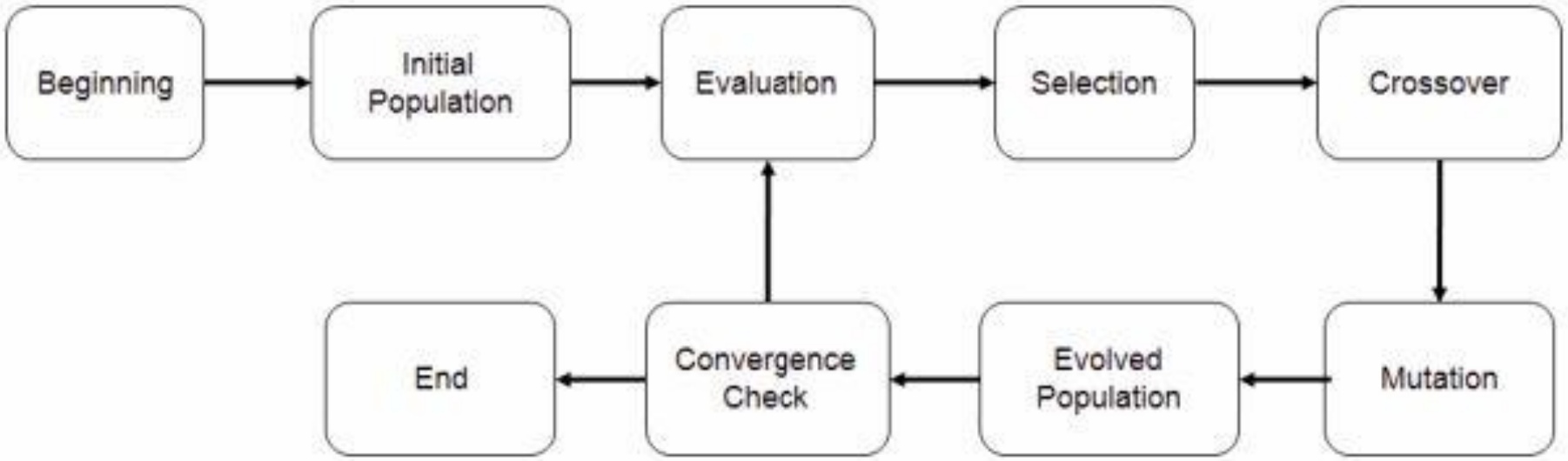
GAs & Biology



GAs: The „Reproduction“ Cycle



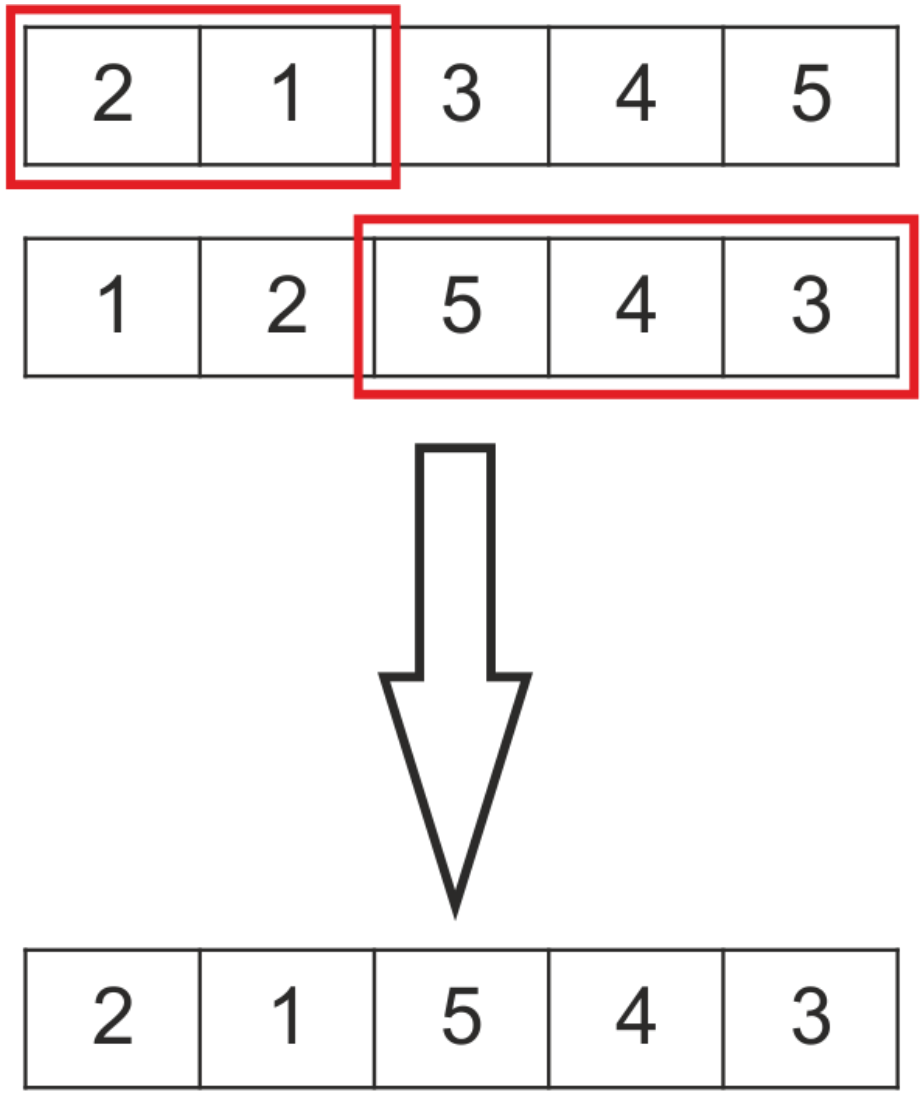
GAs: The „Reproduction“ Cycle



GAs: Main Steps

- Initialization
  - Population
- **Selection**
  - Fitness Function
- **Crossover**
- **Mutation**

Crossover



One-Point Crossover

1	**2**	**3**	**4**	**5**

5	**4**	**3**	**2**	**1**

One-Point Crossover

1	2	3	4	5

5	4	3	2	1

Let's ***i=3***

One-Point Crossover

1	2	3	4	5

5	4	3	2	1

Let's ***i=3***

1	2	3		

5	4	3		

One-Point Crossover

1	2	3	4	5

5	4	3	2	1

Let's ***i=3***

1	2	3	_	

5	4	3		

One-Point Crossover

1	2	3	4	5

5	4	3	2	1

Let's ***i=3***

1	2	3	5	

5	4	3		

One-Point Crossover

1	2	3	4	5

5	4	3	2	1

Let's ***i=3***

1	2	3	5	_

5	4	3		

One-Point Crossover

1	2	3	4	5

5	4	3	2	1

Let's ***i=3***

1	2	3	5	4

5	4	3		

One-Point Crossover

1	2	3	4	5

5	4	3	2	1

Let's ***i=3***

1	2	3	5	4

5	4	3		

One-Point Crossover

1	2	3	4	5

5	4	3	2	1

Let's ***i=3***

1	2	3	5	4

5	4	3	_	

One-Point Crossover

1	2	3	4	5

5	4	3	2	1

Let's ***i=3***

1	2	3	5	4

5	4	3	1	

One-Point Crossover

1	2	3	4	5

5	4	3	2	1

Let's ***i=3***

1	2	3	5	4

5	4	3	1	_

One-Point Crossover

1	2	3	4	5

5	4	3	2	1

Let's ***i=3***

1	2	3	5	4

5	4	3	1	2

One-Point Crossover

1	2	3	4	5

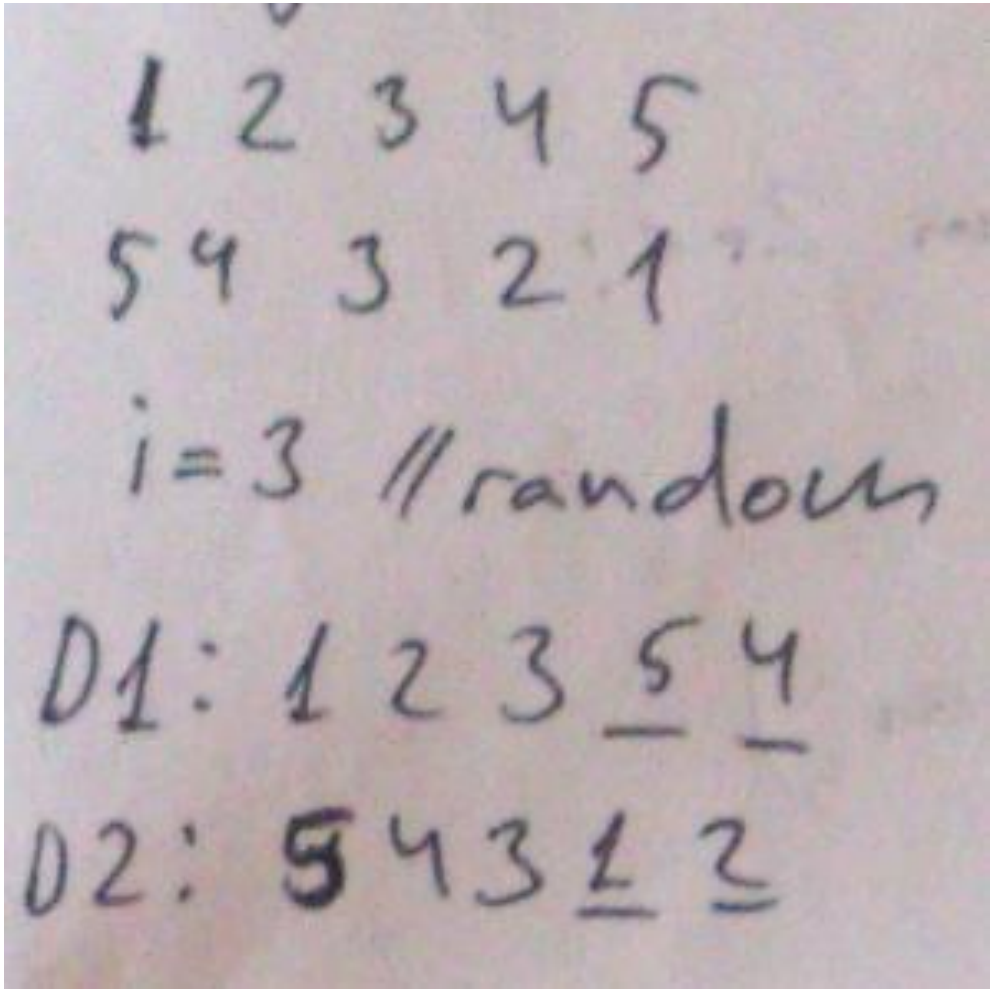
5	4	3	2	1

Let's ***i=3***

1	2	3	5	4

5	4	3	1	2

One-Point Crossover



Two-Point Crossover

1	2	3	4	5	6	7

7	6	2	4	3	5	1

Two-Point Crossover

1	2	3	4	5	6	7

7	6	2	4	3	5	1

Let's ***i=3, j=5***

Two-Point Crossover

1	2	3	4	5	6	7

7	6	2	4	3	5	1

Let's ***i=3, j=5***

		3	4	5		

		2	4	3		

Two-Point Crossover

1	2	3	4	5	6	7

7	6	2	4	3	5	1

Let's ***i=3, j=5***

		3	4	5	_	

		2	4	3		

Two-Point Crossover

1	2	3	4	5	6	7

7	6	2	4	3	5	1

Let's ***i=3, j=5***

		3	4	5	_	

		2	4	3		

Two-Point Crossover

1	2	3	4	5	6	7

7	6	2	4	3	5	1

Let's ***i=3, j=5***

		3	4	5	1	

		2	4	3		

Two-Point Crossover

1	2	3	4	5	6	7

7	6	2	4	3	5	1

Let's ***i=3, j=5***

		3	4	5	1	_

		2	4	3		

Two-Point Crossover

1	2	3	4	5	6	7

7	6	2	4	3	5	1

Let's ***i=3, j=5***

		3	4	5	1	7

		2	4	3		

Two-Point Crossover

1	2	3	4	5	6	7

7	6	2	4	3	5	1

Let's ***i=3, j=5***

_		3	4	5	1	7

		2	4	3		

Two-Point Crossover

1	2	3	4	5	6	7

7	6	2	4	3	5	1

Let's ***i=3, j=5***

6		3	4	5	1	7

		2	4	3		

Two-Point Crossover

1	2	3	4	5	6	7

7	6	2	4	3	5	1

Let's ***i=3, j=5***

6	_	3	4	5	1	7

		2	4	3		

Two-Point Crossover

1	2	3	4	5	6	7

7	6	2	4	3	5	1

Let's ***i=3, j=5***

6	2	3	4	5	1	7

		2	4	3		

Two-Point Crossover

1	2	3	4	5	6	7

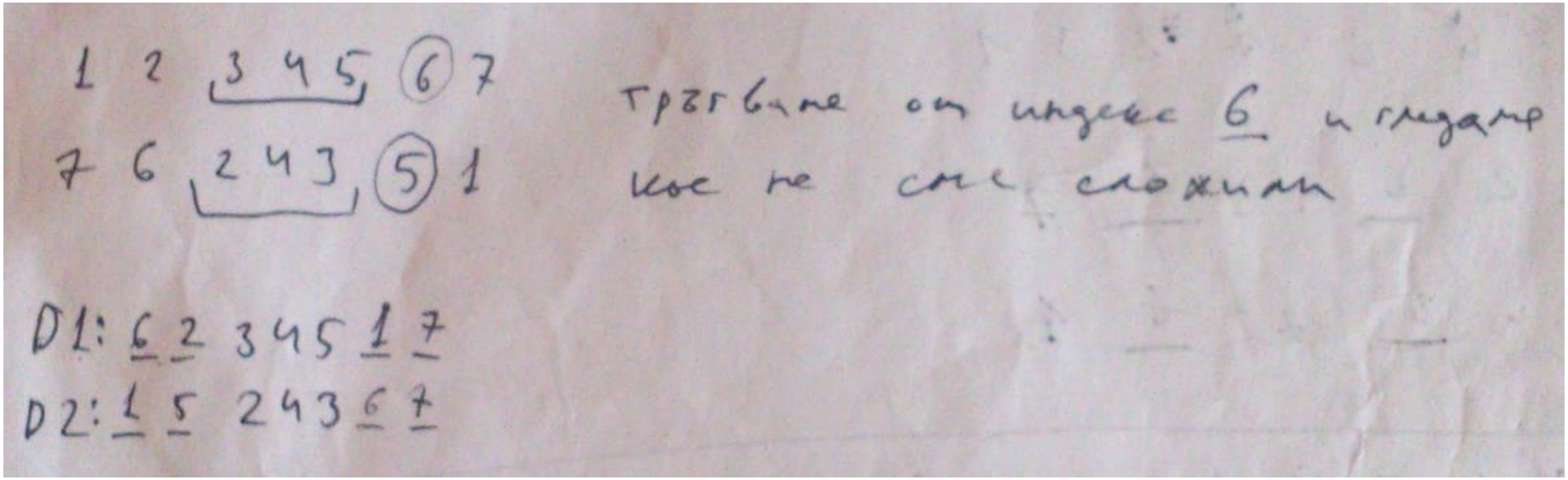
7	6	2	4	3	5	1

Let's ***i=3, j=5***

6	2	3	4	5	1	7

		2	4	3		

Two-Point Crossover



Partially-mapped Crossover

1	2	3	4	5	6	7

6	3	4	1	2	7	5

Partially-mapped Crossover

1	2	3	4	5	6	7

6	3	4	1	2	7	5

Let's ***i=3, j=5***

Partially-mapped Crossover

1	2	3	4	5	6	7

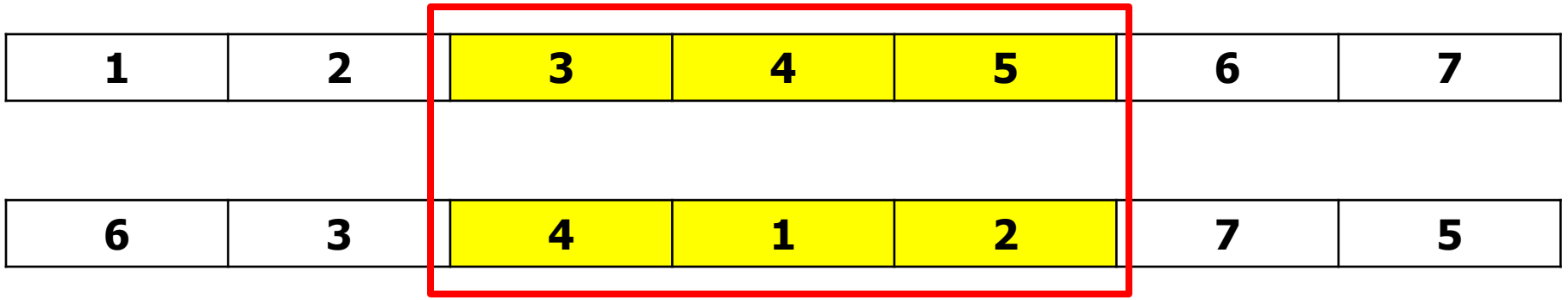
6	3	4	1	2	7	5

Let's ***i=3, j=5***

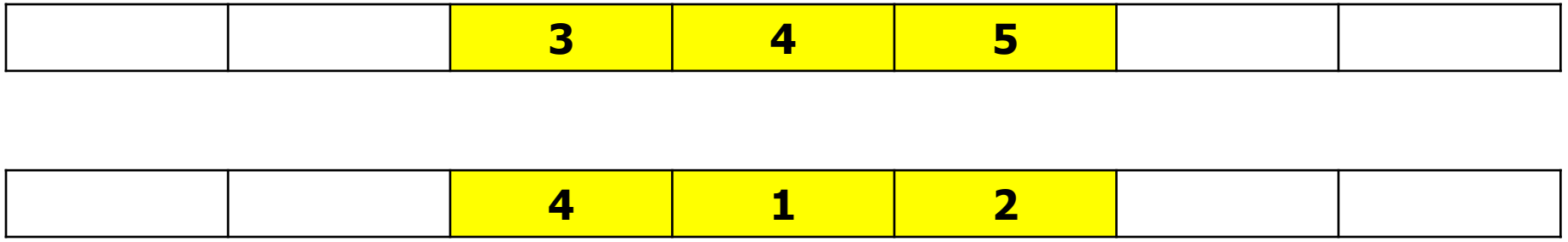
		3	4	5		

		4	1	2		

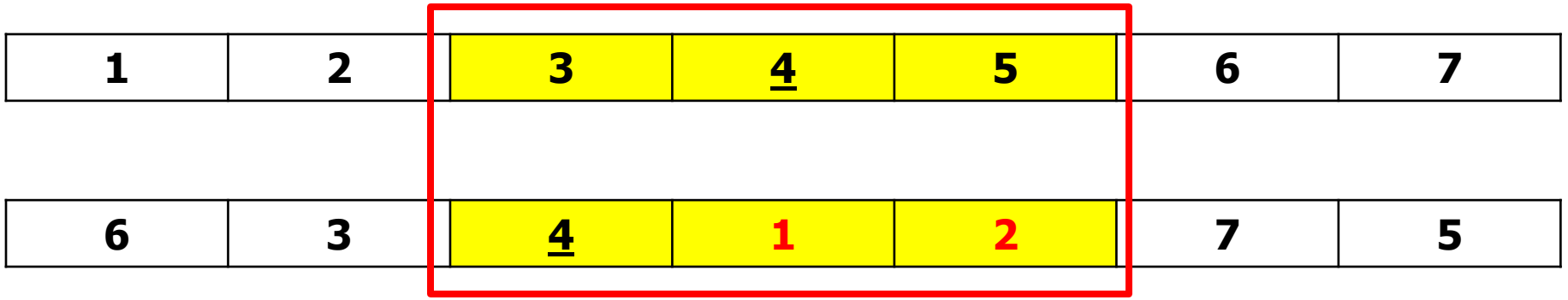
Partially-mapped Crossover



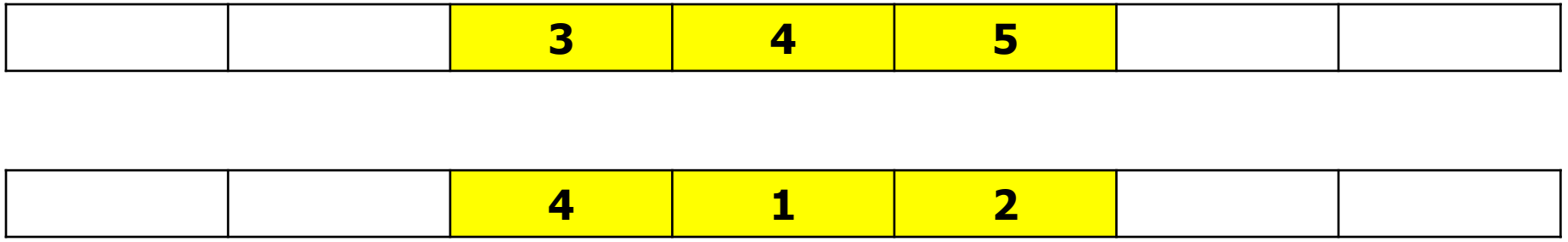
Let's ***i=3, j=5***



Partially-mapped Crossover



Let's ***i=3, j=5***



Partially-mapped Crossover

1	2	3	4	5	6	7

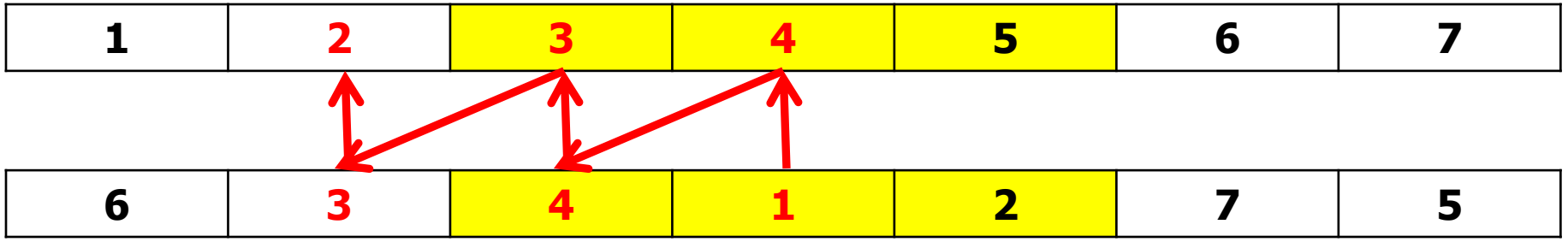
6	3	4	1	2	7	5

Let's ***i=3, j=5***

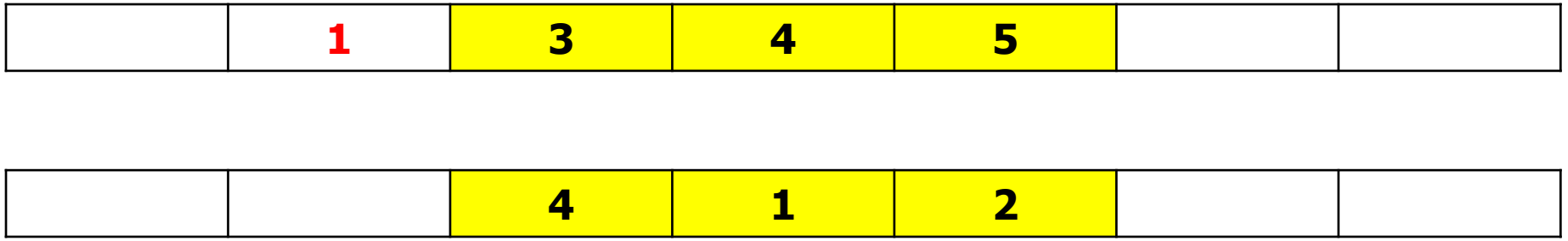
		3	4	5		

		4	1	2		

Partially-mapped Crossover



Let's ***i=3, j=5***



Partially-mapped Crossover

1	2	3	4	5	6	7

6	3	4	1	2	7	5

Let's ***i=3, j=5***

	1	3	4	5		

		4	1	2		

Partially-mapped Crossover

1	2	3	4	5	6	7

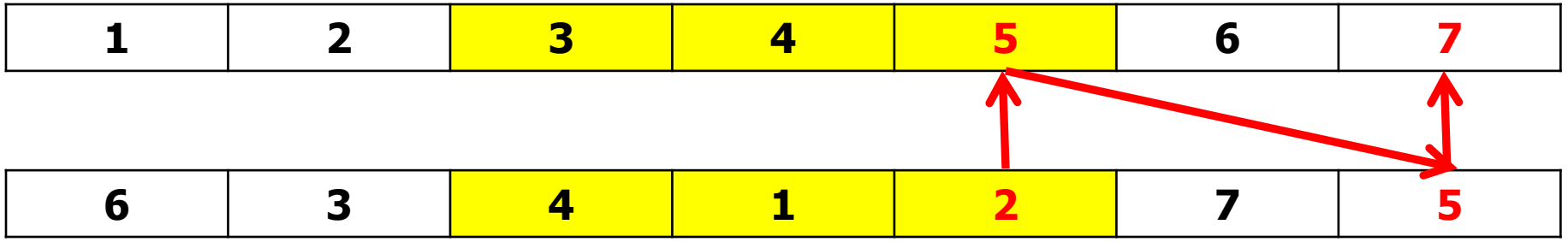
6	3	4	1	2	7	5

Let's ***i=3, j=5***

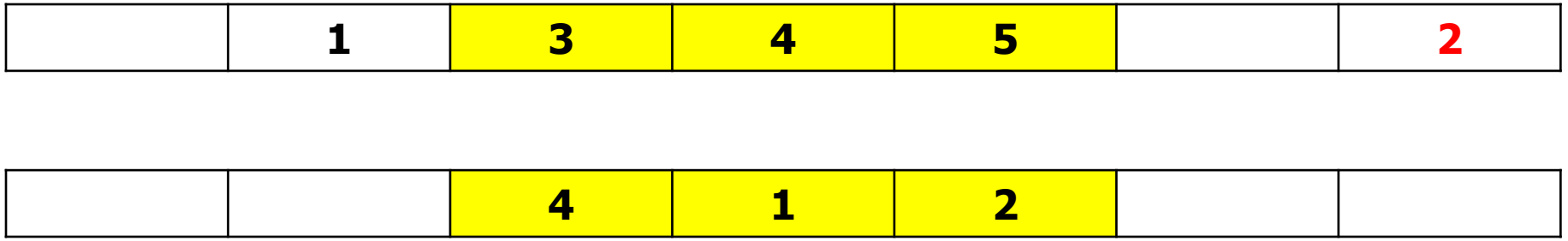
	1	3	4	5		

		4	1	2		

Partially-mapped Crossover



Let's ***i=3, j=5***



Partially-mapped Crossover

1	2	3	4	5	6	7

6	3	4	1	2	7	5

Let's ***i=3, j=5***

	1	3	4	5		2

		4	1	2		

Partially-mapped Crossover

1	2	3	4	5	6	7

6	3	4	1	2	7	5

Let's ***i=3, j=5***

_	1	3	4	5	_	2

		4	1	2		

Partially-mapped Crossover

1	2	3	4	5	6	7

6	3	4	1	2	7	5

Let's ***i=3, j=5***

6	1	3	4	5	7	2

		4	1	2		

Partially-mapped Crossover

1	2	3	4	5	6	7

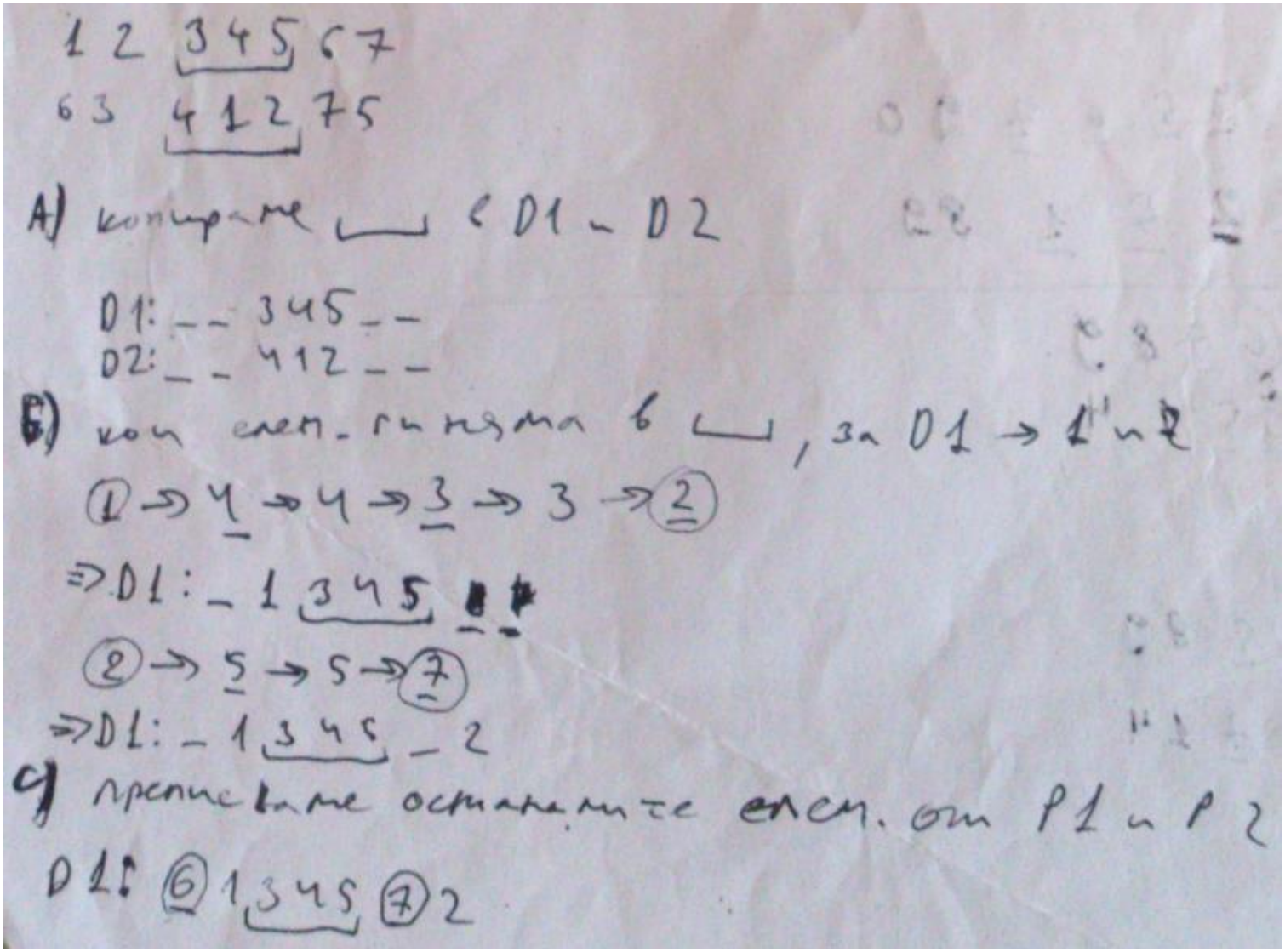
6	3	4	1	2	7	5

Let's ***i=3, j=5***

6	1	3	4	5	7	2

		4	1	2		

Partially-mapped Crossover



Cycle Crossover

8	4	7	3	6	2	5	1	9	0

0	1	2	3	4	5	6	7	8	9

Cycle Crossover

8	**4**	**7**	**3**	**6**	**2**	**5**	**1**	**9**	**0**

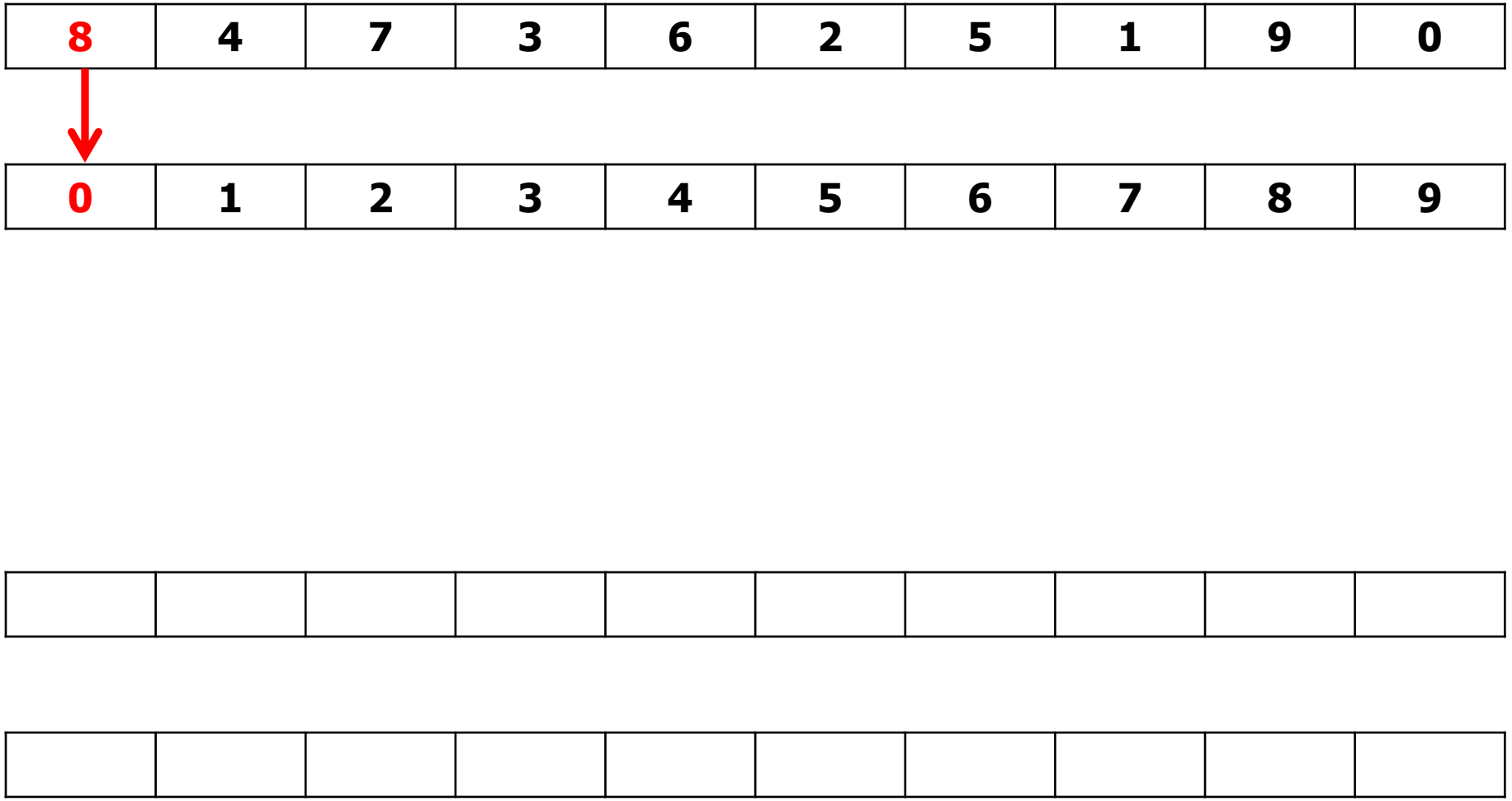
0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

Cycle Crossover

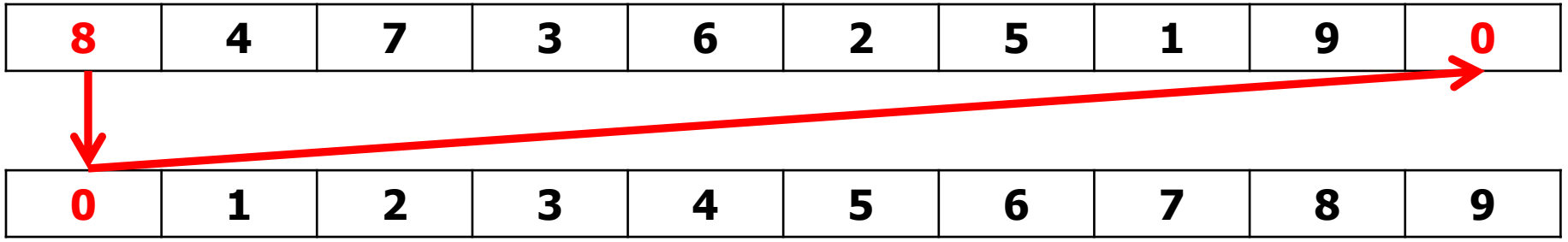
8	**4**	**7**	**3**	**6**	**2**	**5**	**1**	**9**	**0**

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

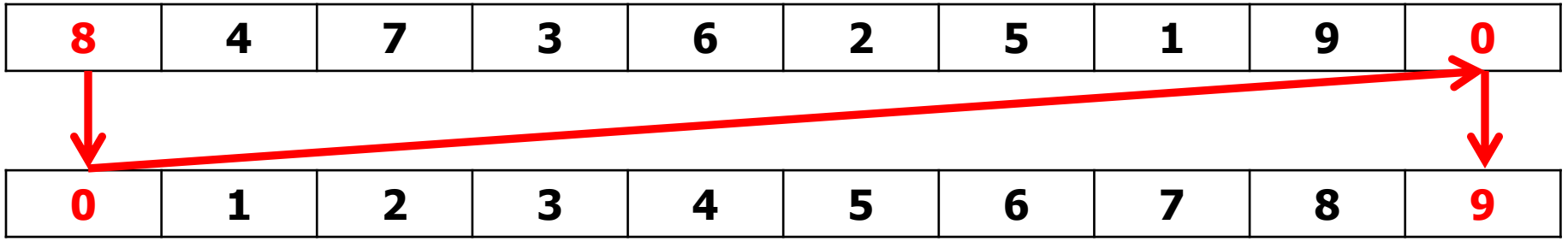
Cycle Crossover



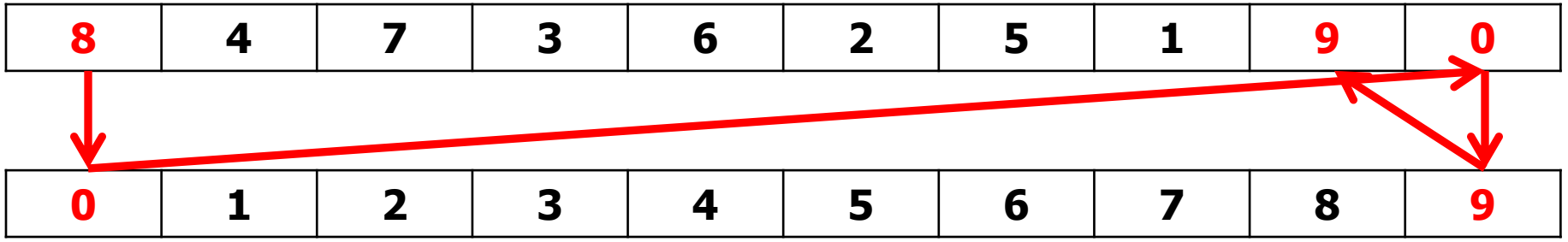
Cycle Crossover



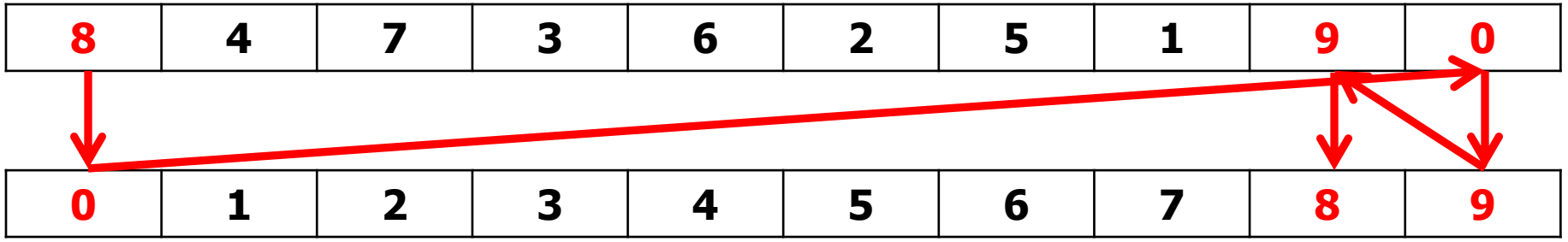
Cycle Crossover



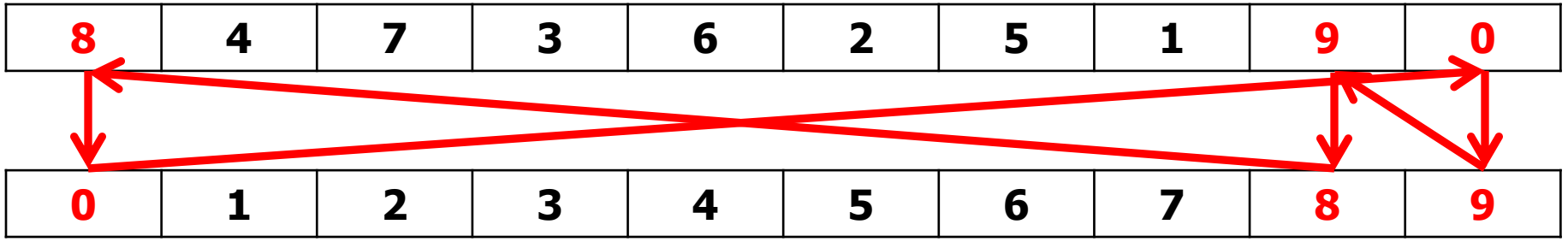
Cycle Crossover



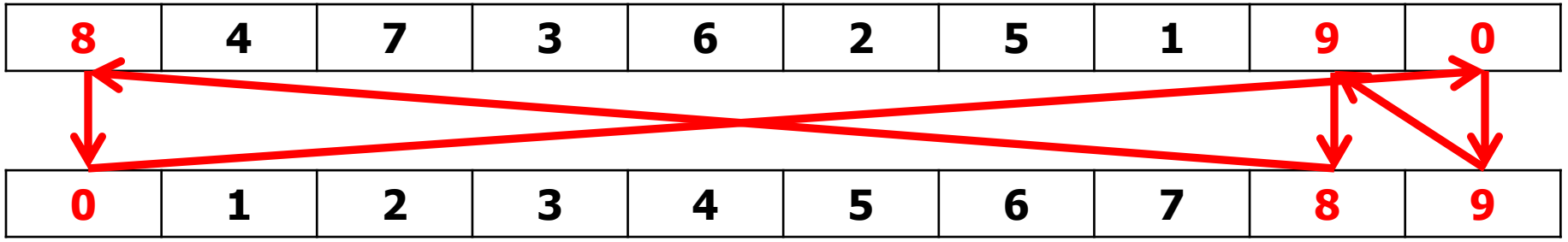
Cycle Crossover



Cycle Crossover



Cycle Crossover



8								9	0

0								8	9

Cycle Crossover

8	**4**	**7**	**3**	**6**	**2**	**5**	**1**	**9**	**0**

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

8								**9**	**0**

0								**8**	**9**

Cycle Crossover

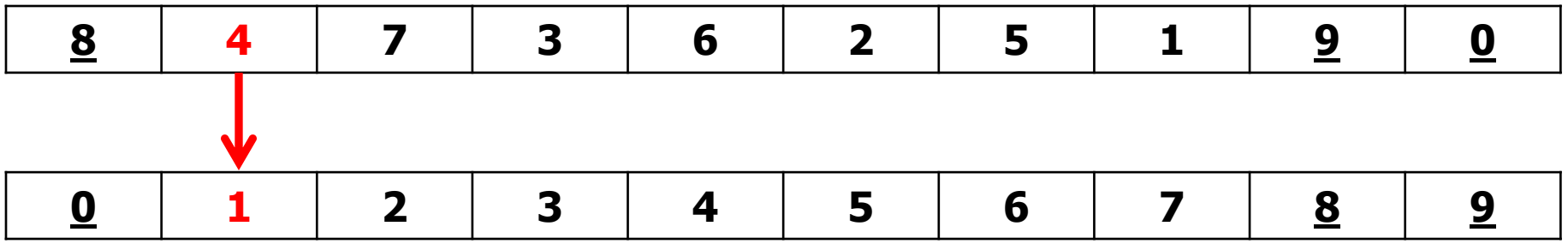
8	**4**	**7**	**3**	**6**	**2**	**5**	**1**	**9**	**0**

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

8								**9**	**0**

0								**8**	**9**

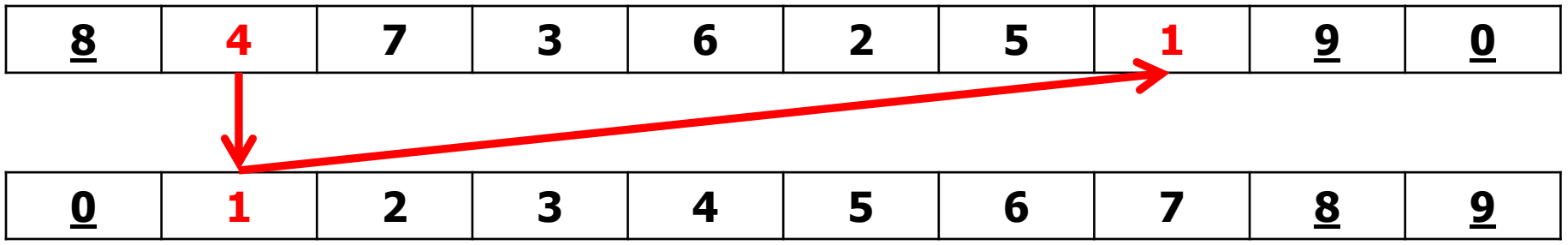
Cycle Crossover



8								9	0

0								8	9

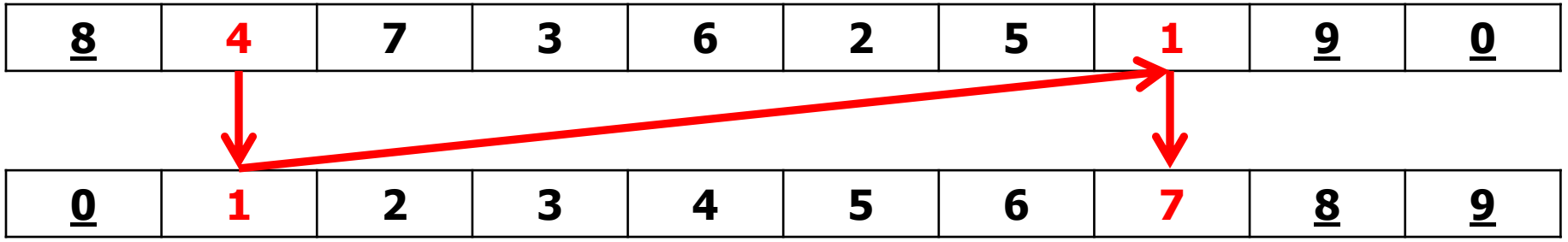
Cycle Crossover



8								9	0

0								8	9

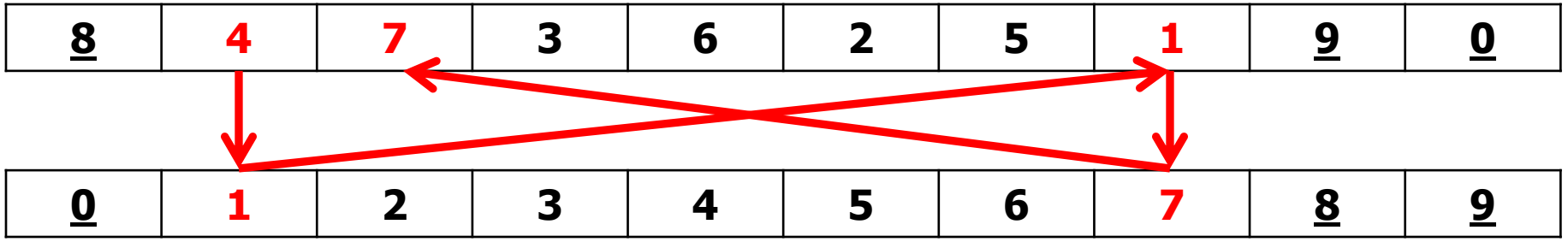
Cycle Crossover



8								9	0

0								8	9

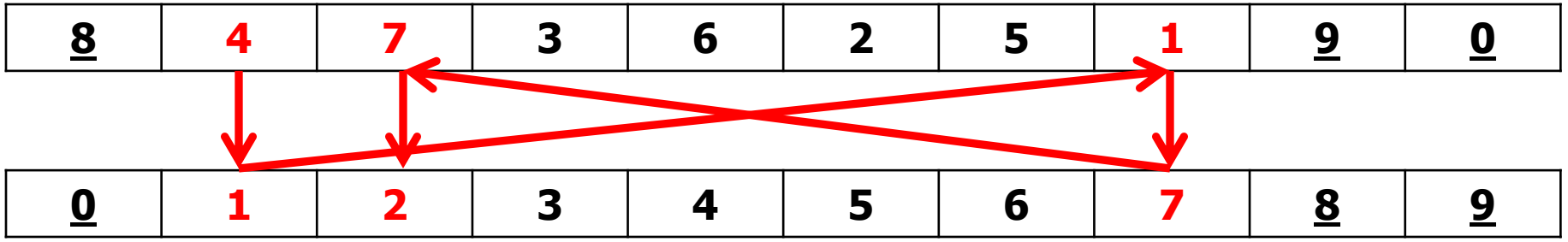
Cycle Crossover



8								9	0

0								8	9

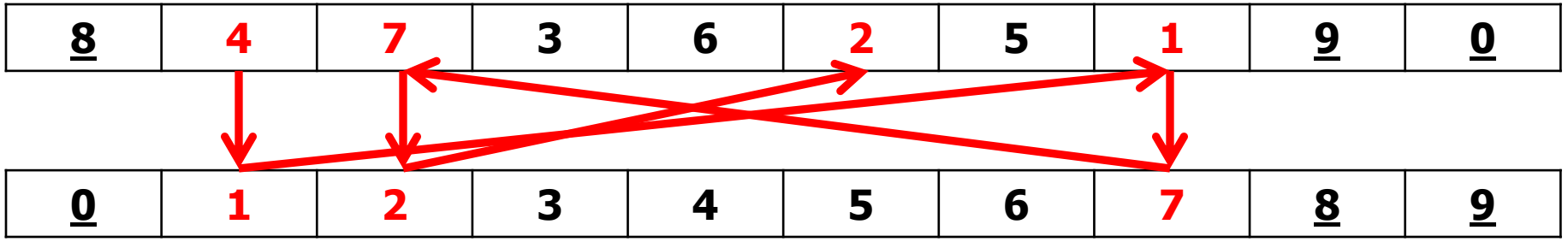
Cycle Crossover



8								9	0

0								8	9

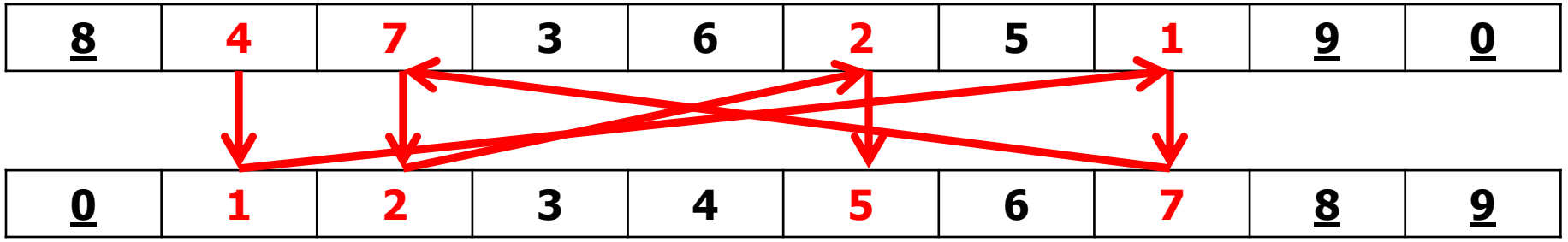
Cycle Crossover



8								9	0

0								8	9

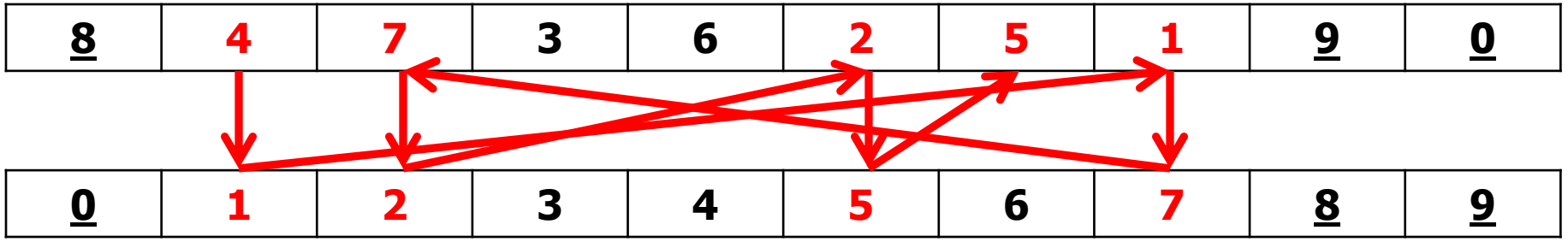
Cycle Crossover



8								9	0

0								8	9

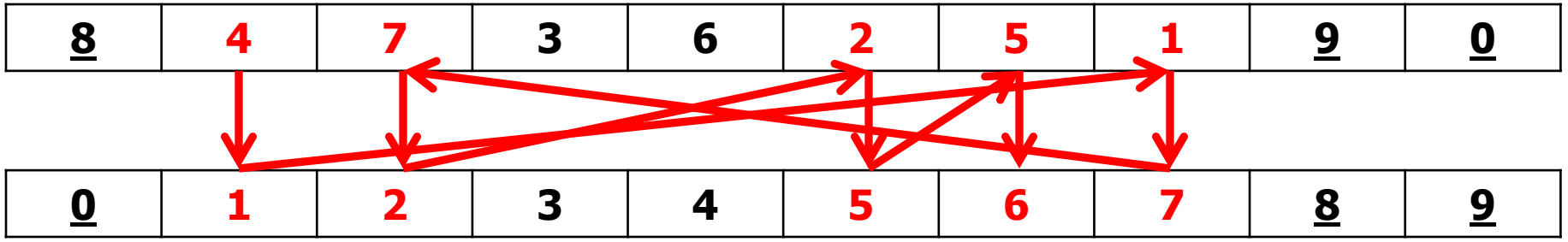
Cycle Crossover



8								9	0

0								8	9

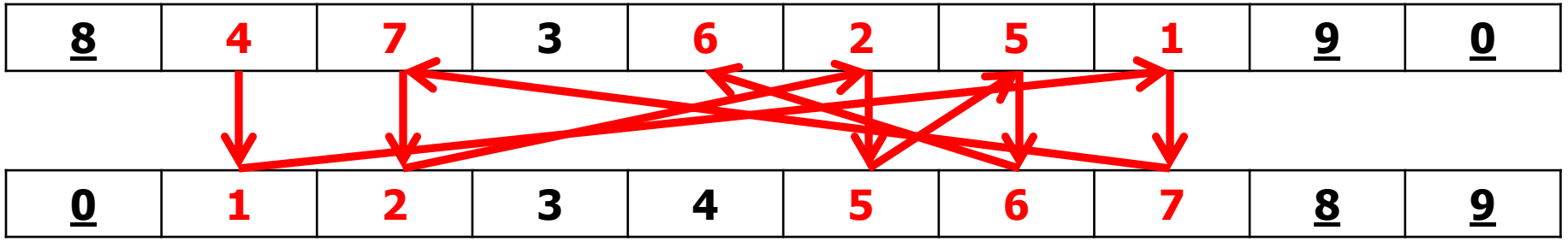
Cycle Crossover



8								9	0

0								8	9

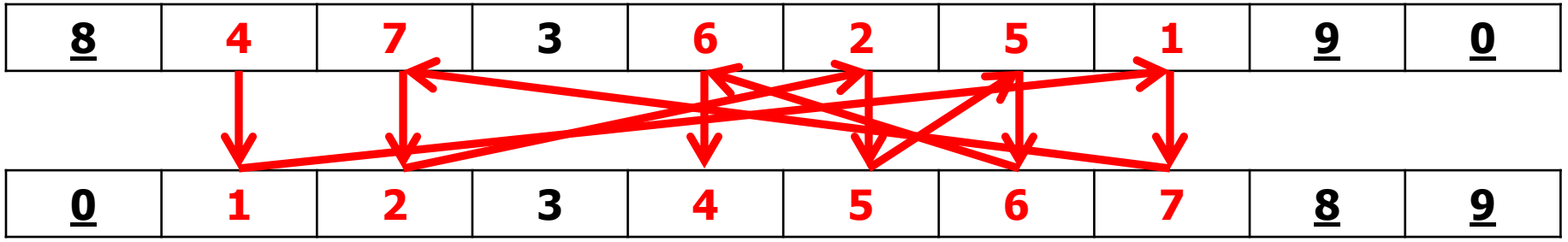
Cycle Crossover



8								9	0

0								8	9

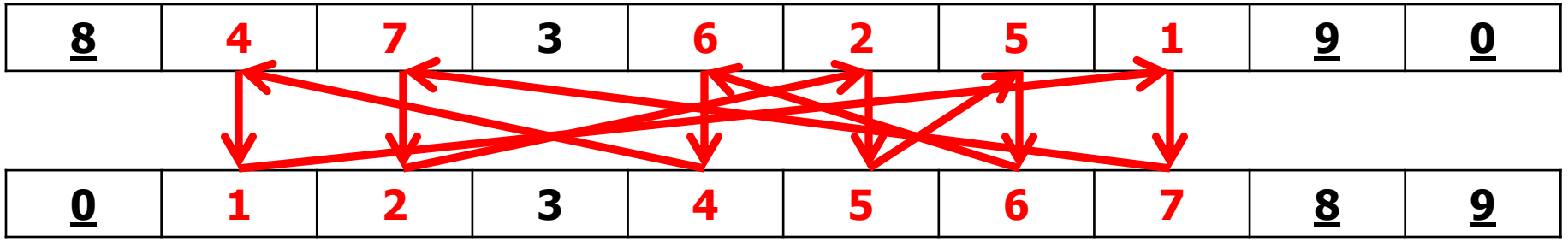
Cycle Crossover



8								9	0

0								8	9

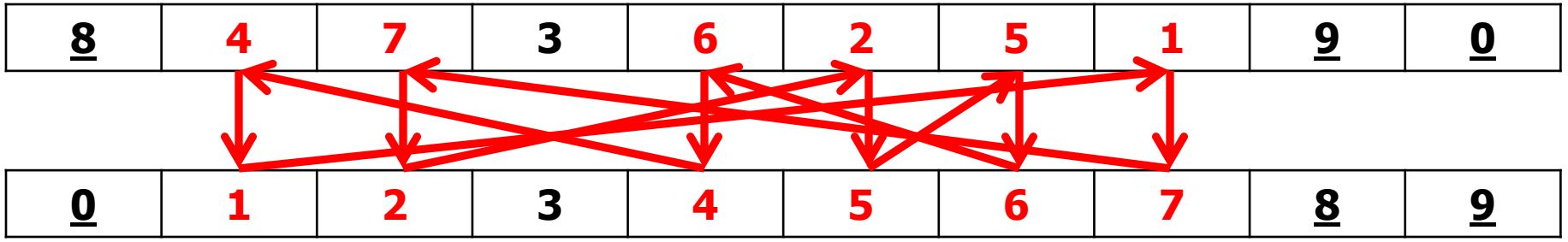
Cycle Crossover



8								9	0

0								8	9

Cycle Crossover



8	1	2		4	5	6	7	9	0

0	4	7		6	2	5	1	8	9

Cycle Crossover

8	**4**	**7**	**3**	**6**	**2**	**5**	**1**	**9**	**0**

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

8	**1**	**2**		**4**	**5**	**6**	**7**	**9**	**0**

0	**4**	**7**		**6**	**2**	**5**	**1**	**8**	**9**

Cycle Crossover

8	**4**	**7**	**3**	**6**	**2**	**5**	**1**	**9**	**0**

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

8	**1**	**2**		**4**	**5**	**6**	**7**	**9**	**0**

0	**4**	**7**		**6**	**2**	**5**	**1**	**8**	**9**

Cycle Crossover

8	4	7	3	6	2	5	1	9	0

0	1	2	3	4	5	6	7	8	9

8	1	2		4	5	6	7	9	0

0	4	7		6	2	5	1	8	9

Cycle Crossover

8	4	7	3	6	2	5	1	9	0

0	1	2	3	4	5	6	7	8	9

8	1	2	3	4	5	6	7	9	0

0	4	7	3	6	2	5	1	8	9

Cycle Crossover

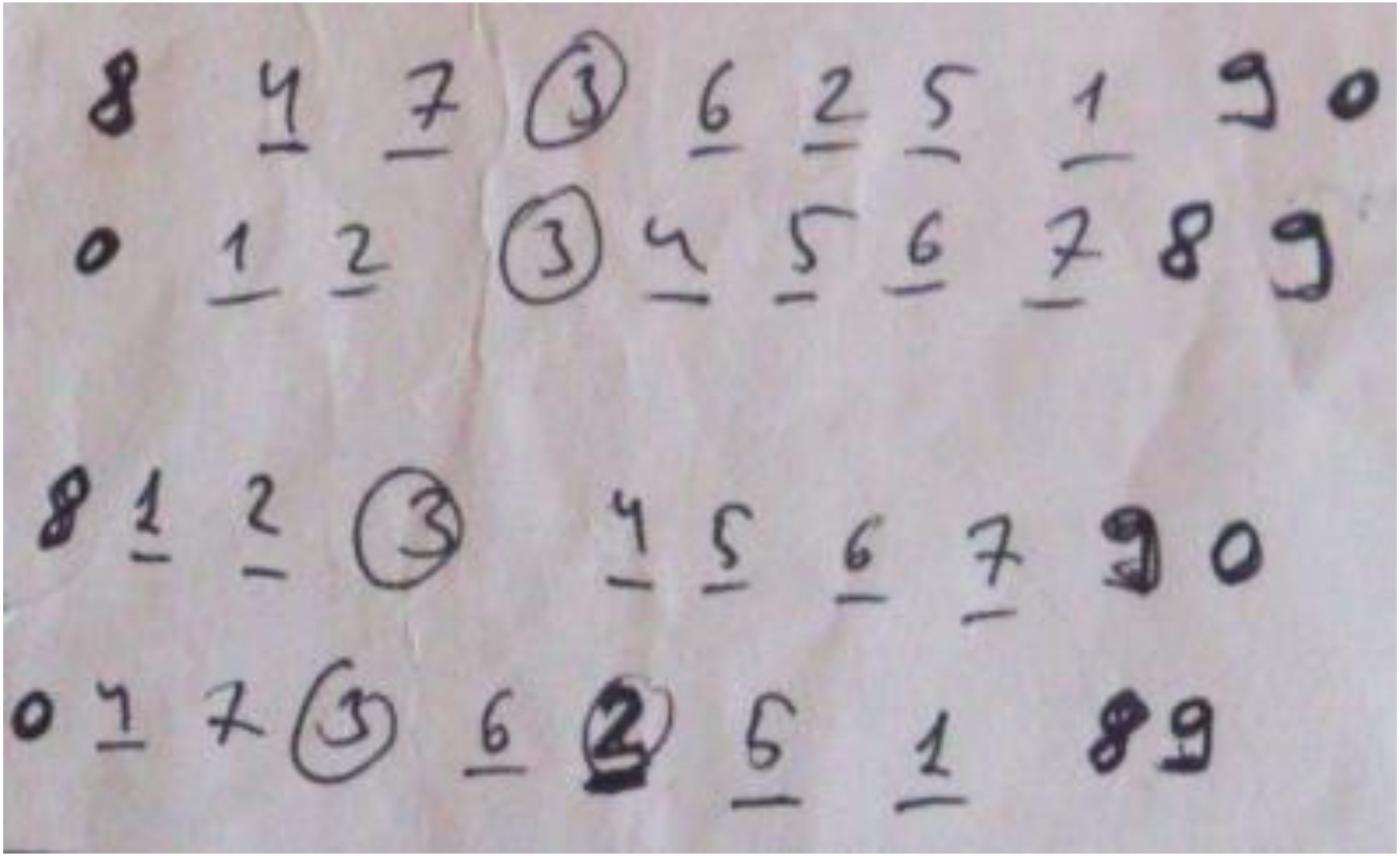
8	**4**	**7**	**3**	**6**	**2**	**5**	**1**	**9**	**0**

0	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**

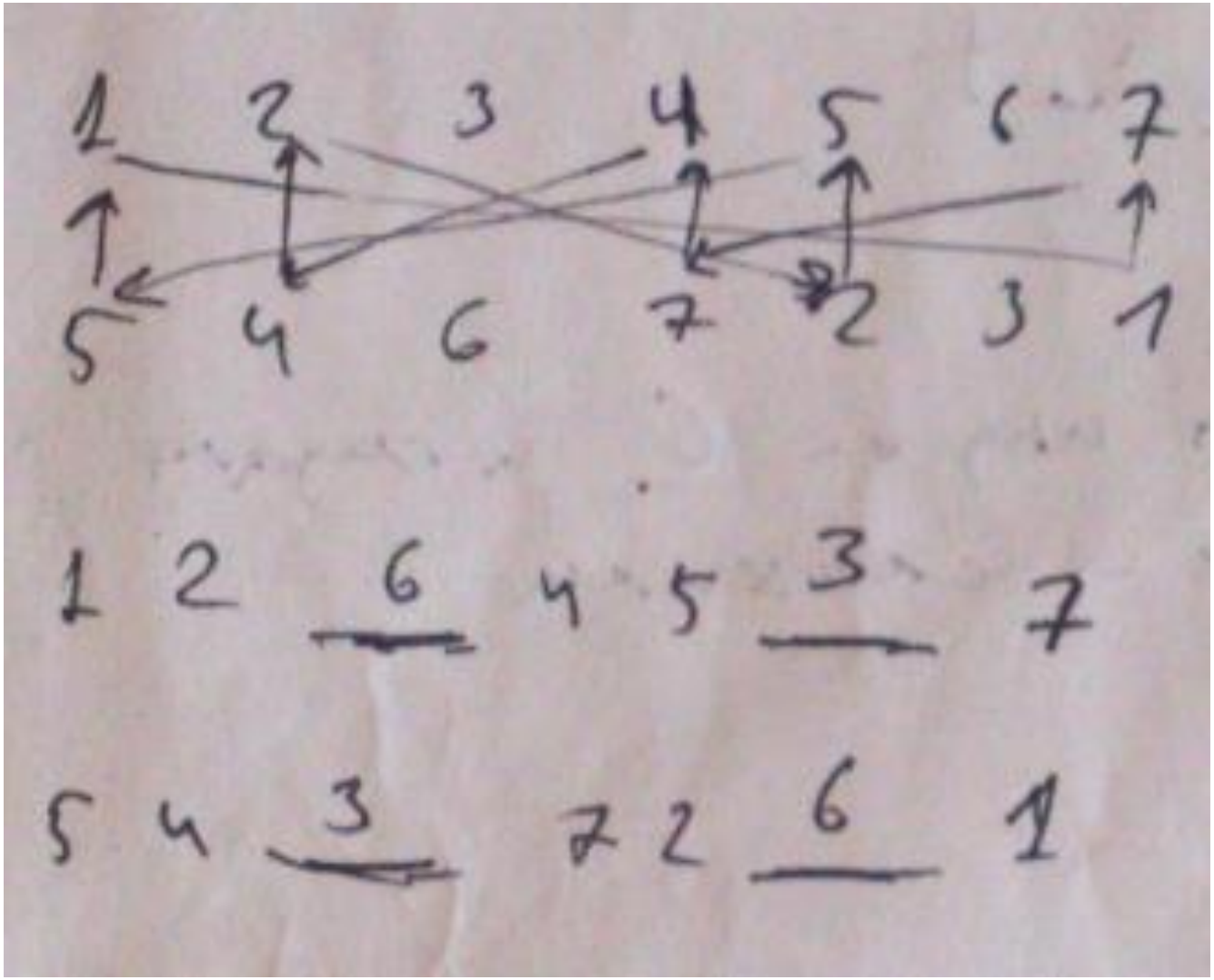
8	**1**	**2**	**3**	**4**	**5**	**6**	**7**	**9**	**0**

0	**4**	**7**	**3**	**6**	**2**	**5**	**1**	**8**	**9**

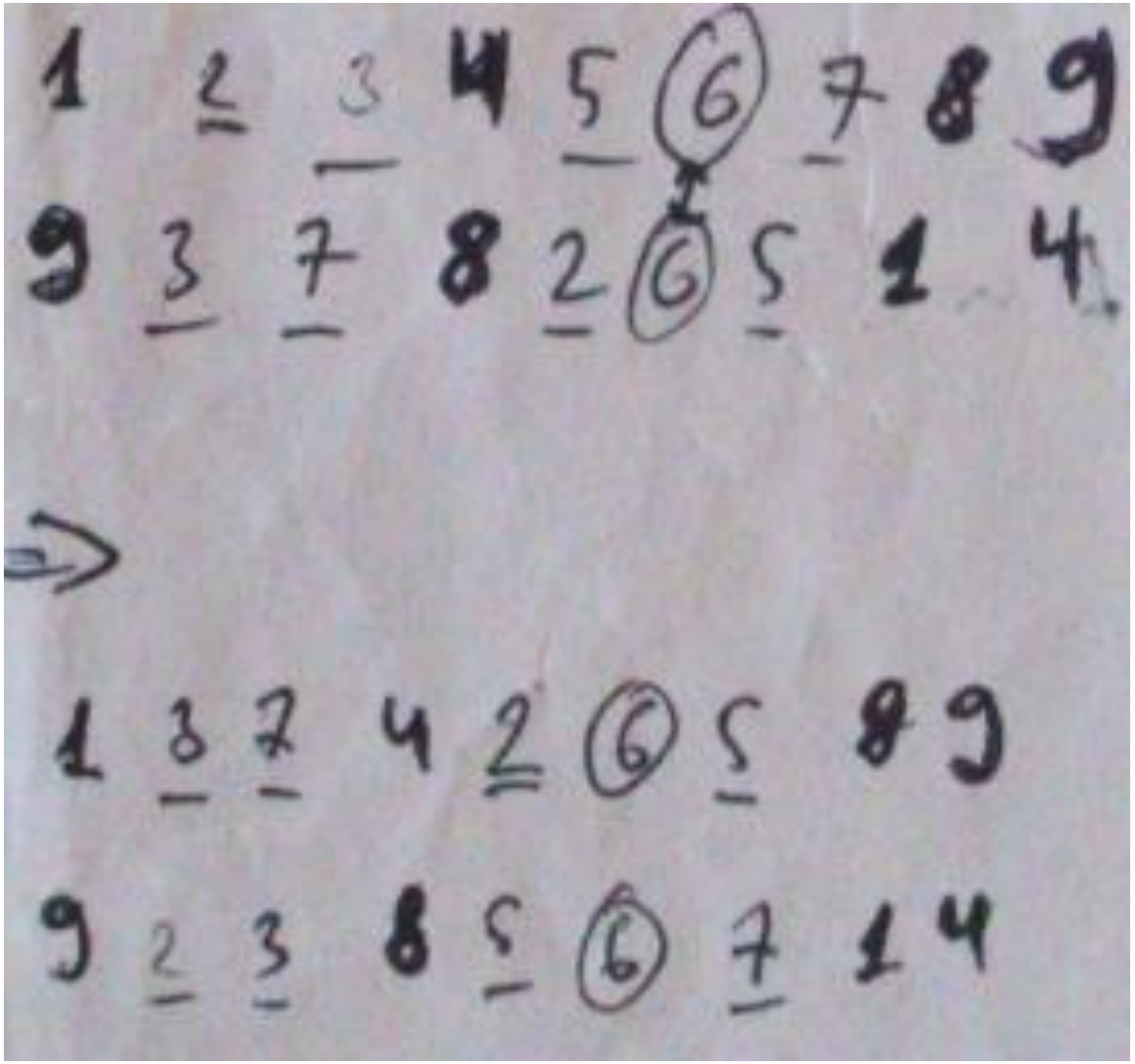
Cycle Crossover



Cycle Crossover



Cycle Crossover



Uniform Crossover

1	2	3	4	5	6	7	8	9	1

2	3	4	5	5	7	7	9	9	0

Uniform Crossover

1	2	3	4	5	6	7	8	9	1

2	3	4	5	5	7	7	9	9	0

Getting a ***gene*** from one of the parents randomly!

Uniform Crossover

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1**

2	**3**	**4**	**5**	**5**	**7**	**7**	**9**	**9**	**0**

Getting a ***gene*** from one of the parents randomly!

Uniform Crossover

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1**

2	**3**	**4**	**5**	**5**	**7**	**7**	**9**	**9**	**0**

Getting a ***gene*** from one of the parents randomly!

2									

Uniform Crossover

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1**

2	**3**	**4**	**5**	**5**	**7**	**7**	**9**	**9**	**0**

Getting a ***gene*** from one of the parents randomly!

2	**2**								

Uniform Crossover

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1**

2	**3**	**4**	**5**	**5**	**7**	**7**	**9**	**9**	**0**

Getting a ***gene*** from one of the parents randomly!

2	**2**	**4**							

Uniform Crossover

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1**

2	**3**	**4**	**5**	**5**	**7**	**7**	**9**	**9**	**0**

Getting a ***gene*** from one of the parents randomly!

2	**2**	**4**	**4**						

Uniform Crossover

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1**

2	**3**	**4**	**5**	**5**	**7**	**7**	**9**	**9**	**0**

Getting a ***gene*** from one of the parents randomly!

2	**2**	**4**	**4**	**5**					

Uniform Crossover

1	2	3	4	5	6	7	8	9	1

2	3	4	5	5	7	7	9	9	0

Getting a ***gene*** from one of the parents randomly!

2	2	4	4	5	7				

Uniform Crossover

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1**

2	**3**	**4**	**5**	**5**	**7**	**7**	**9**	**9**	**0**

Getting a ***gene*** from one of the parents randomly!

2	**2**	**4**	**4**	**5**	**7**	**7**			

Uniform Crossover

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1**

2	**3**	**4**	**5**	**5**	**7**	**7**	**9**	**9**	**0**

Getting a ***gene*** from one of the parents randomly!

2	**2**	**4**	**4**	**5**	**7**	**7**	**9**		

Uniform Crossover

1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1**

2	**3**	**4**	**5**	**5**	**7**	**7**	**9**	**9**	**0**

Getting a ***gene*** from one of the parents randomly!

2	**2**	**4**	**4**	**5**	**7**	**7**	**9**	**9**	

Uniform Crossover

1	2	3	4	5	6	7	8	9	1

2	3	4	5	5	7	7	9	9	0

Getting a ***gene*** from one of the parents randomly!

2	2	4	4	5	7	7	9	9	0

Uniform Crossover

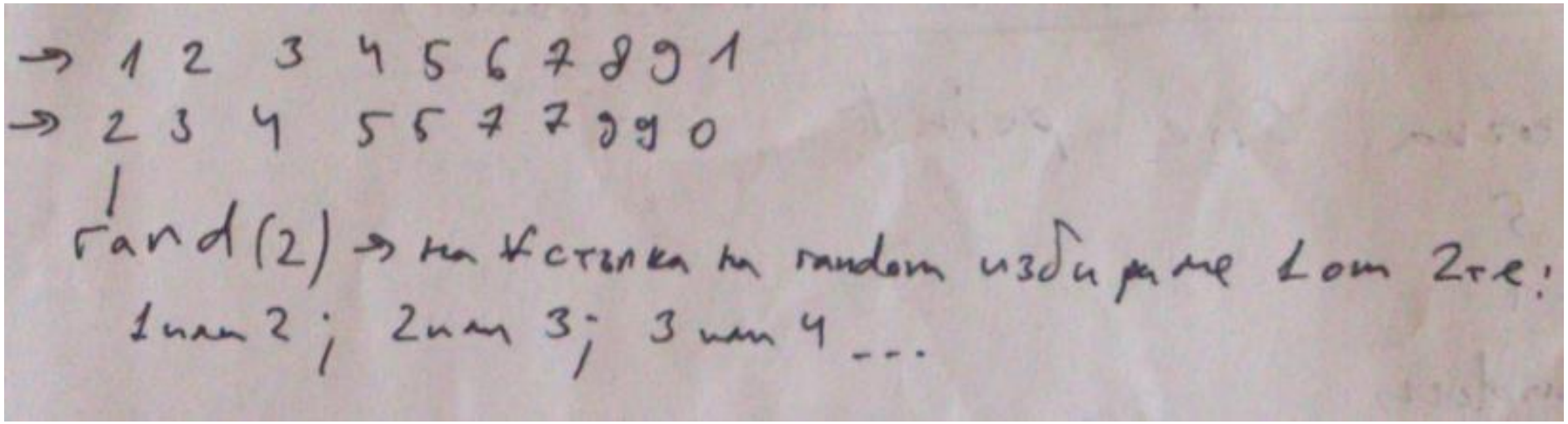
1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**1**

2	**3**	**4**	**5**	**5**	**7**	**7**	**9**	**9**	**0**

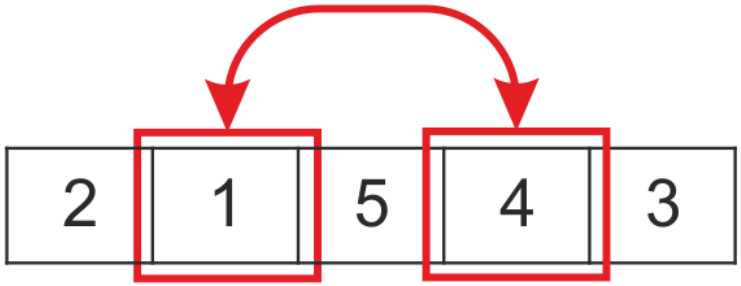
Getting a ***gene*** from one of the parents randomly!

2	**2**	**4**	**4**	**5**	**7**	**7**	**9**	**9**	**0**

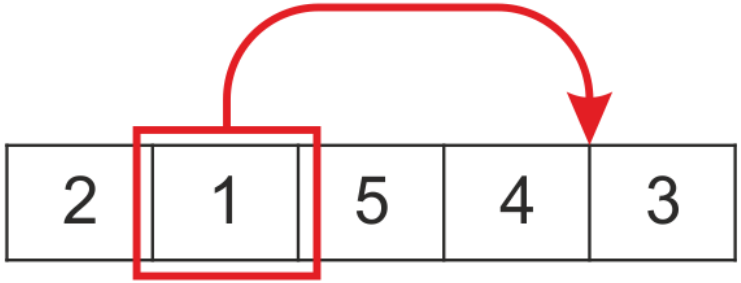
Uniform Crossover



Mutation



swap



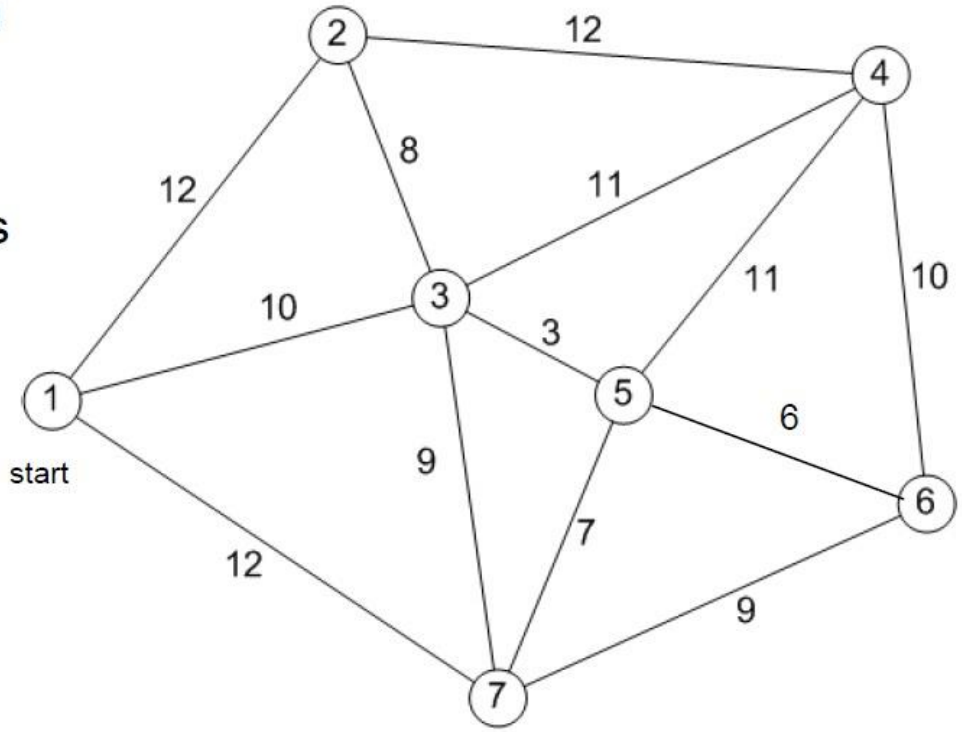
insertion

reverse

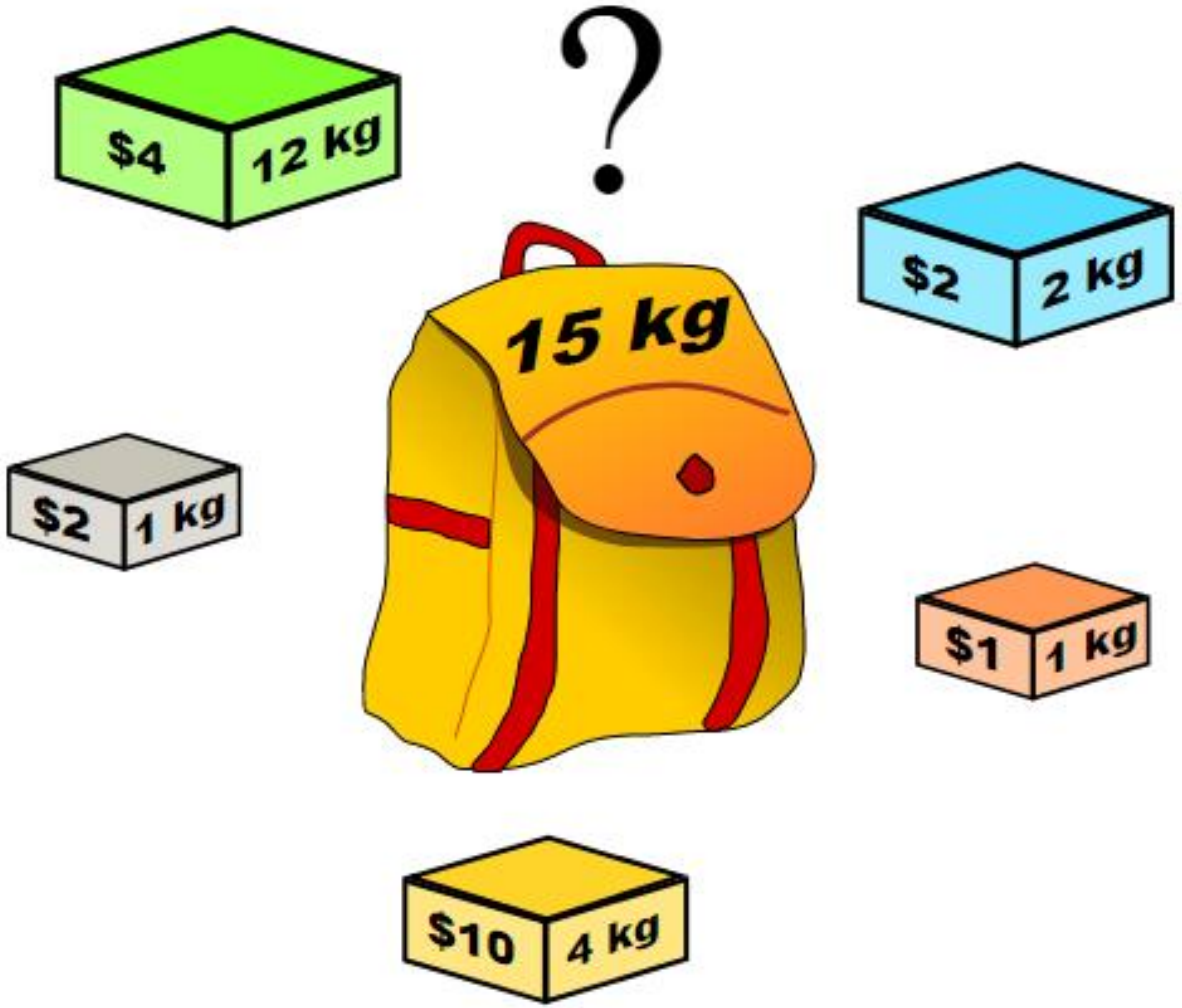
Travelling salesman problem

The Traveling Salesman Problem

- Starting from city 1, the salesman must travel to all cities once before returning home
- The distance between each city is given, and is assumed to be the same in both directions
- Only the links shown are to be used
- Objective - Minimize the total distance to be travelled



Knapsack problem



```
t = 0;
initialize(P(t=0));
evaluate(P(t=0));
while isNotTerminated() do
    P_p(t) = P(t).selectParents();
    P_c(t) = reproduction(P_p);
    mutate(P_c(t));
    evaluate(P_c(t));
    P(t+1) = buildNextGenerationFrom(P_c(t), P(t));
    t = t +1;
end
```