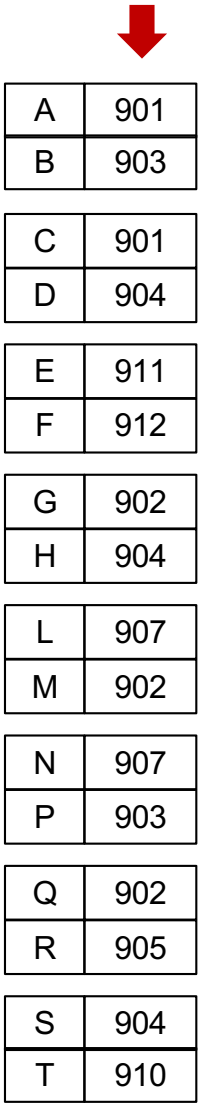
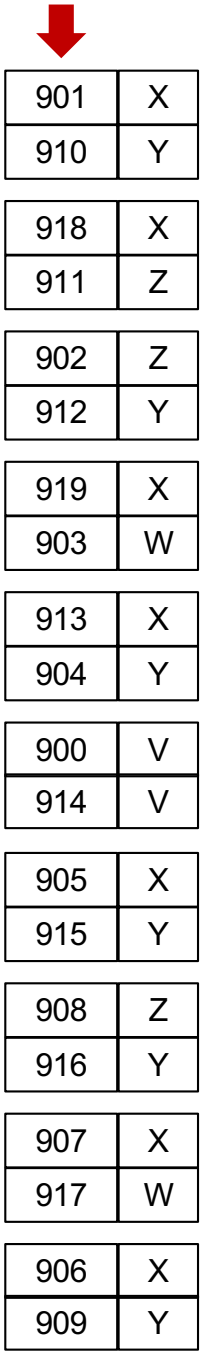
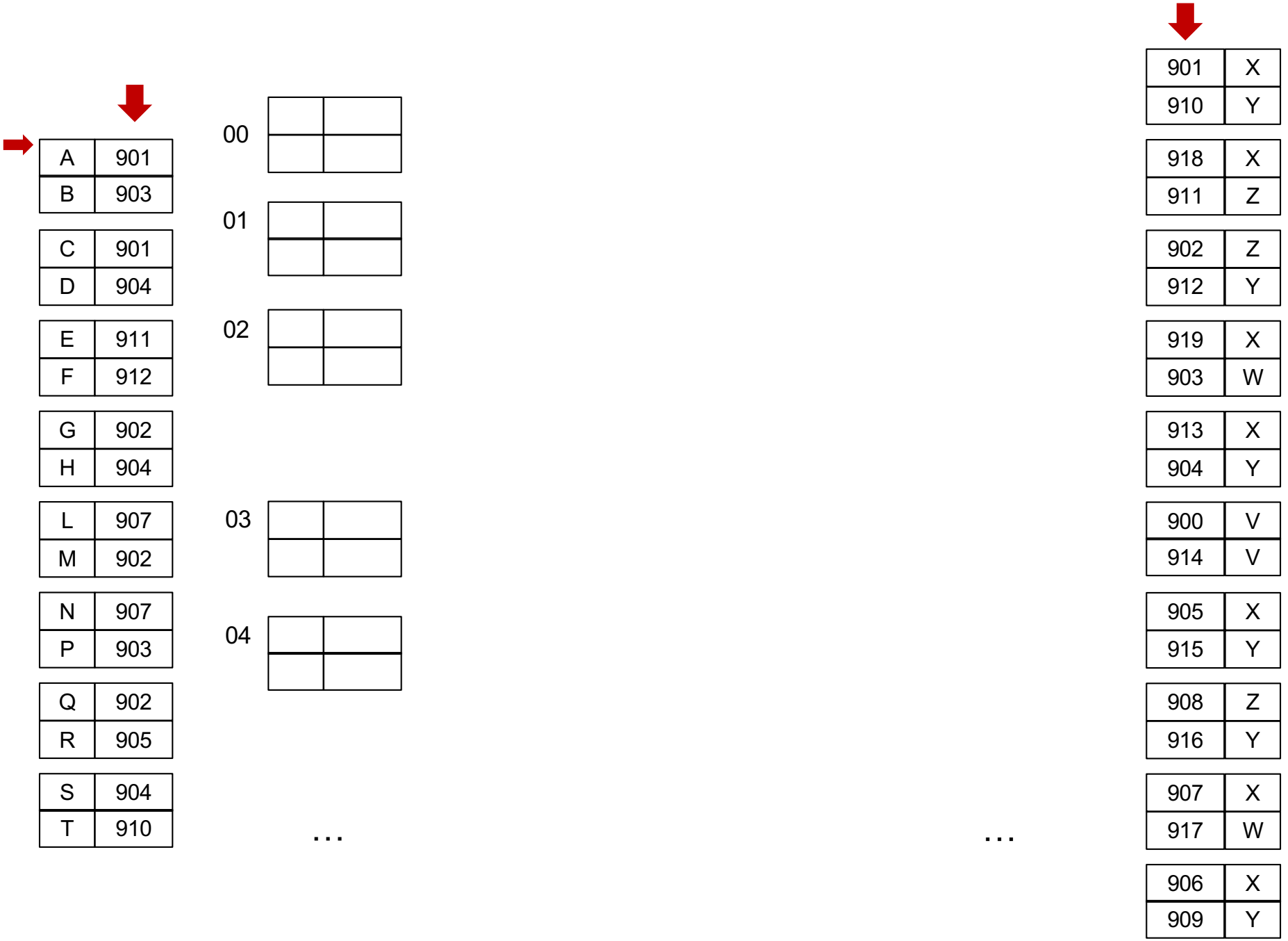
Hash join

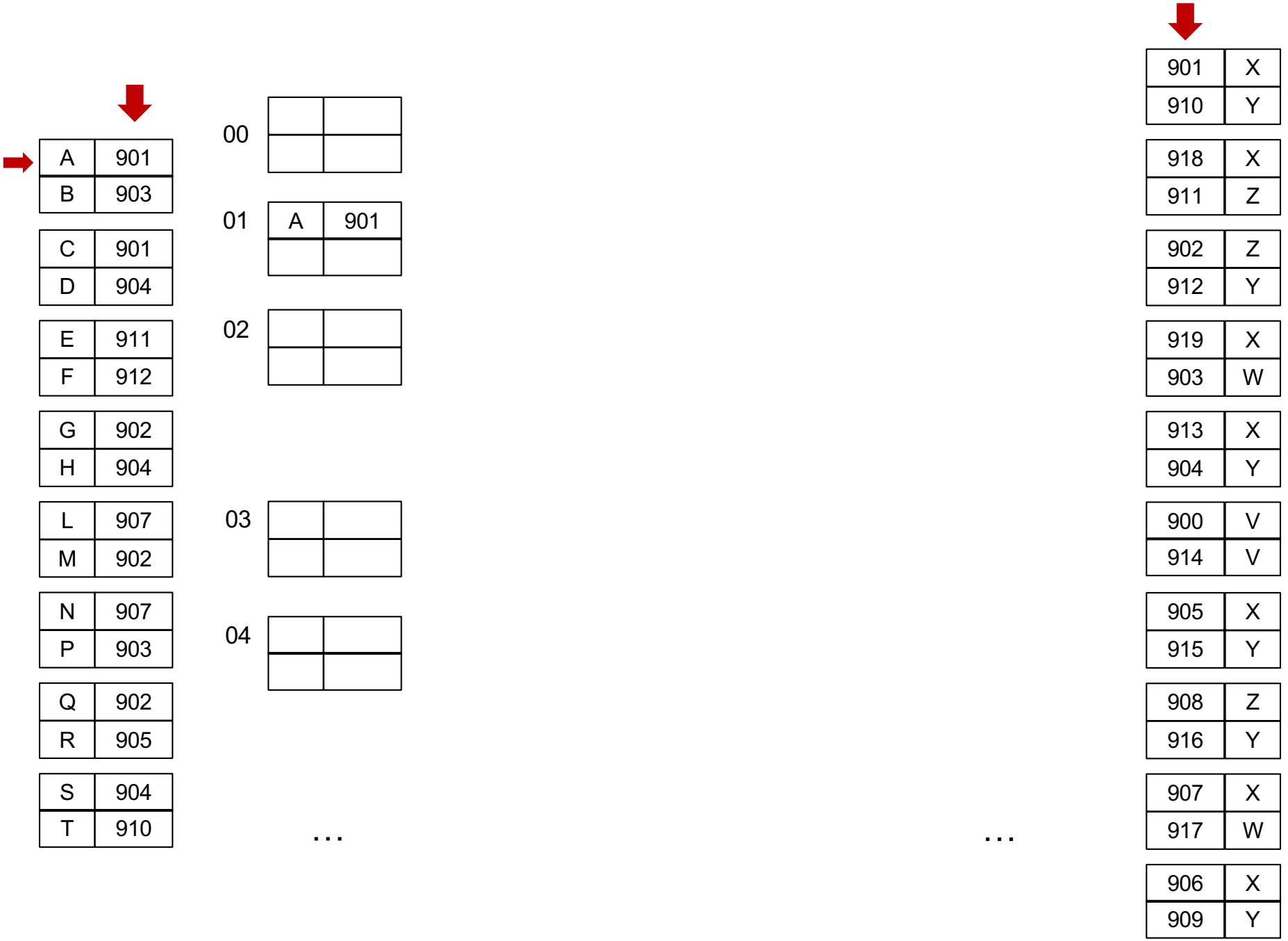


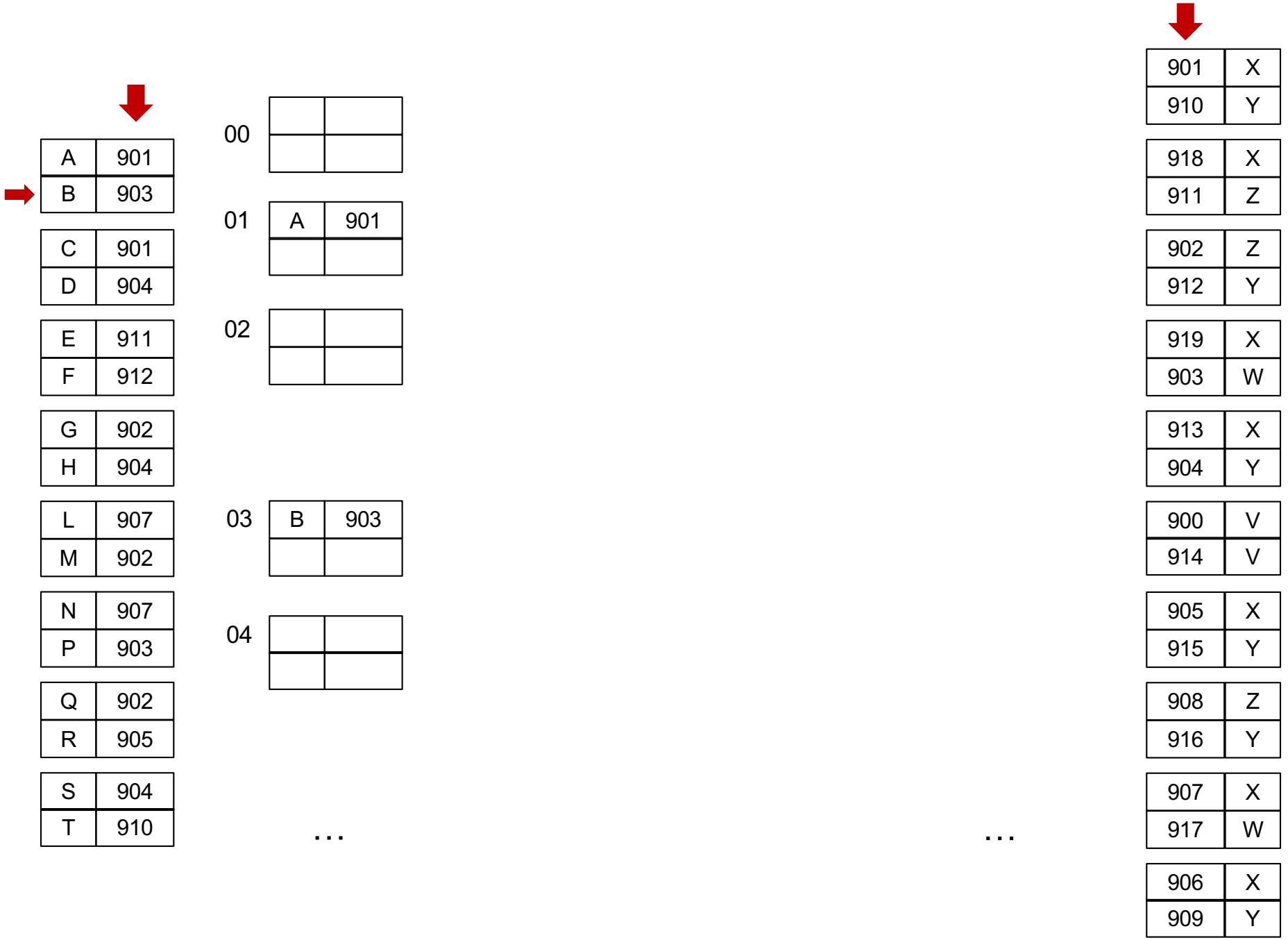
A	901
B	903
C	901
D	904
E	911
F	912
G	902
H	904
L	907
M	902
N	907
P	903
Q	902
R	905
S	904
T	910

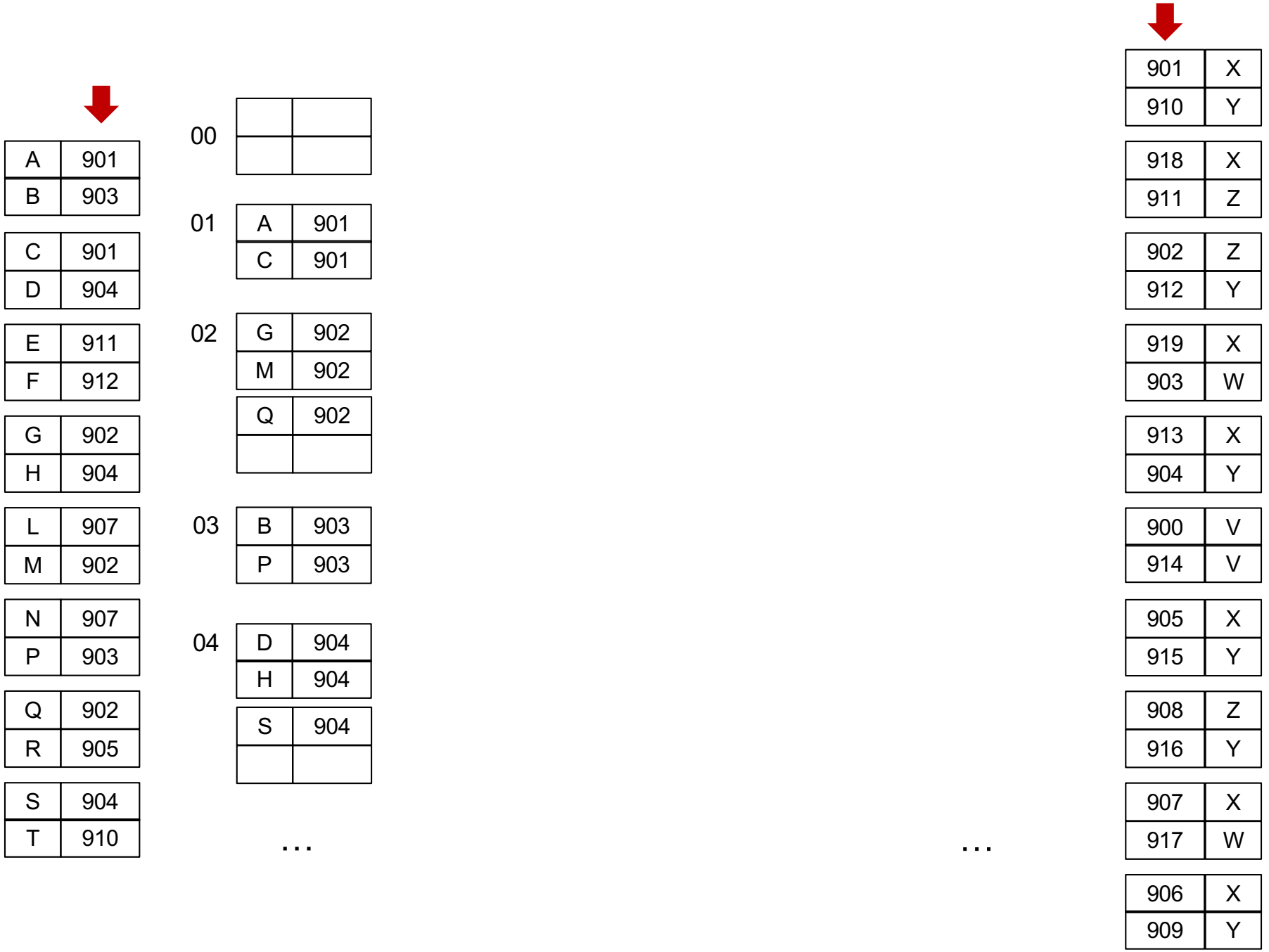


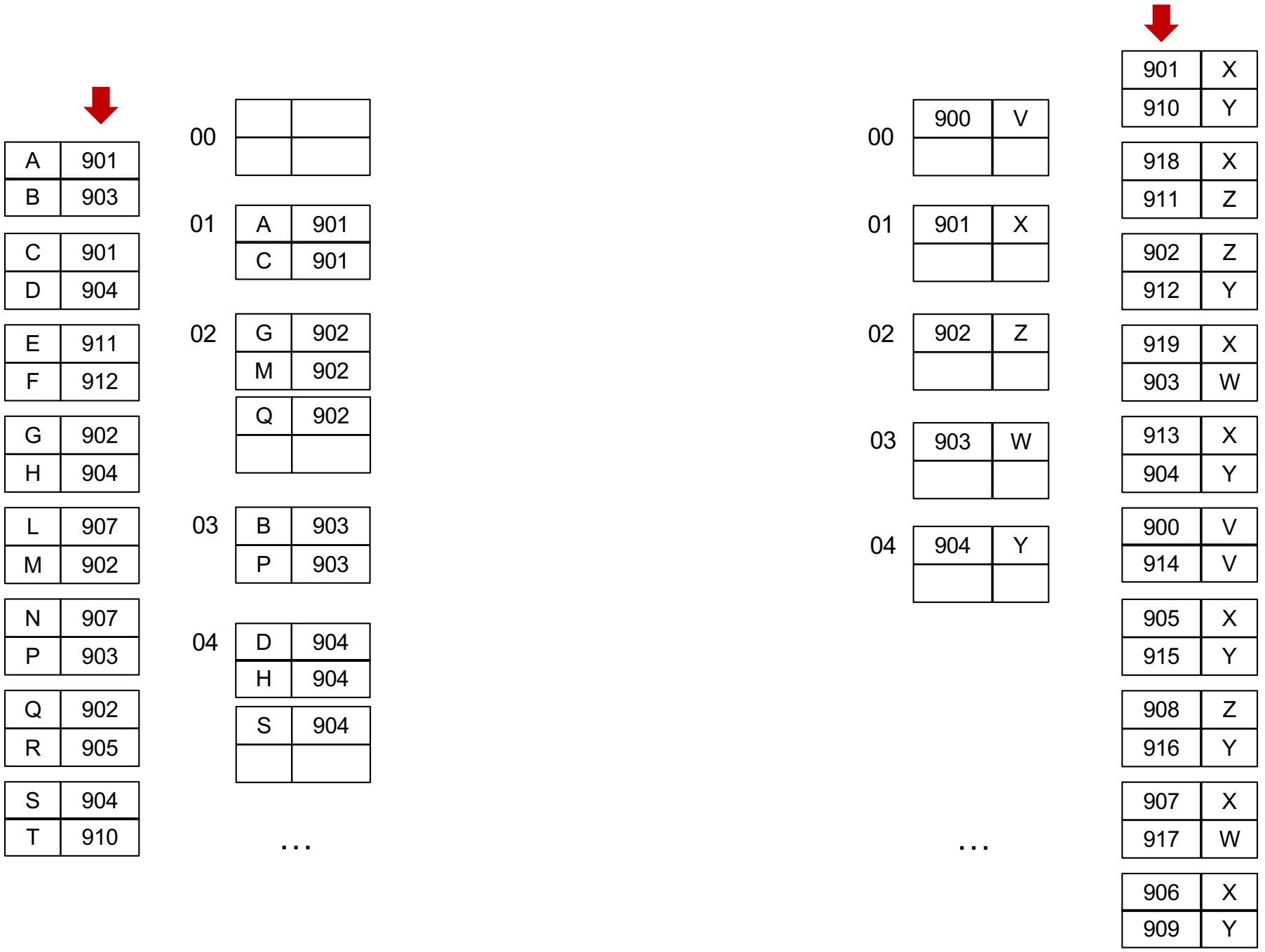
901	X
910	Y
918	X
911	Z
902	Z
912	Y
919	X
903	W
913	X
904	Y
900	V
914	V
905	X
915	Y
908	Z
916	Y
907	X
917	W
906	X
909	Y

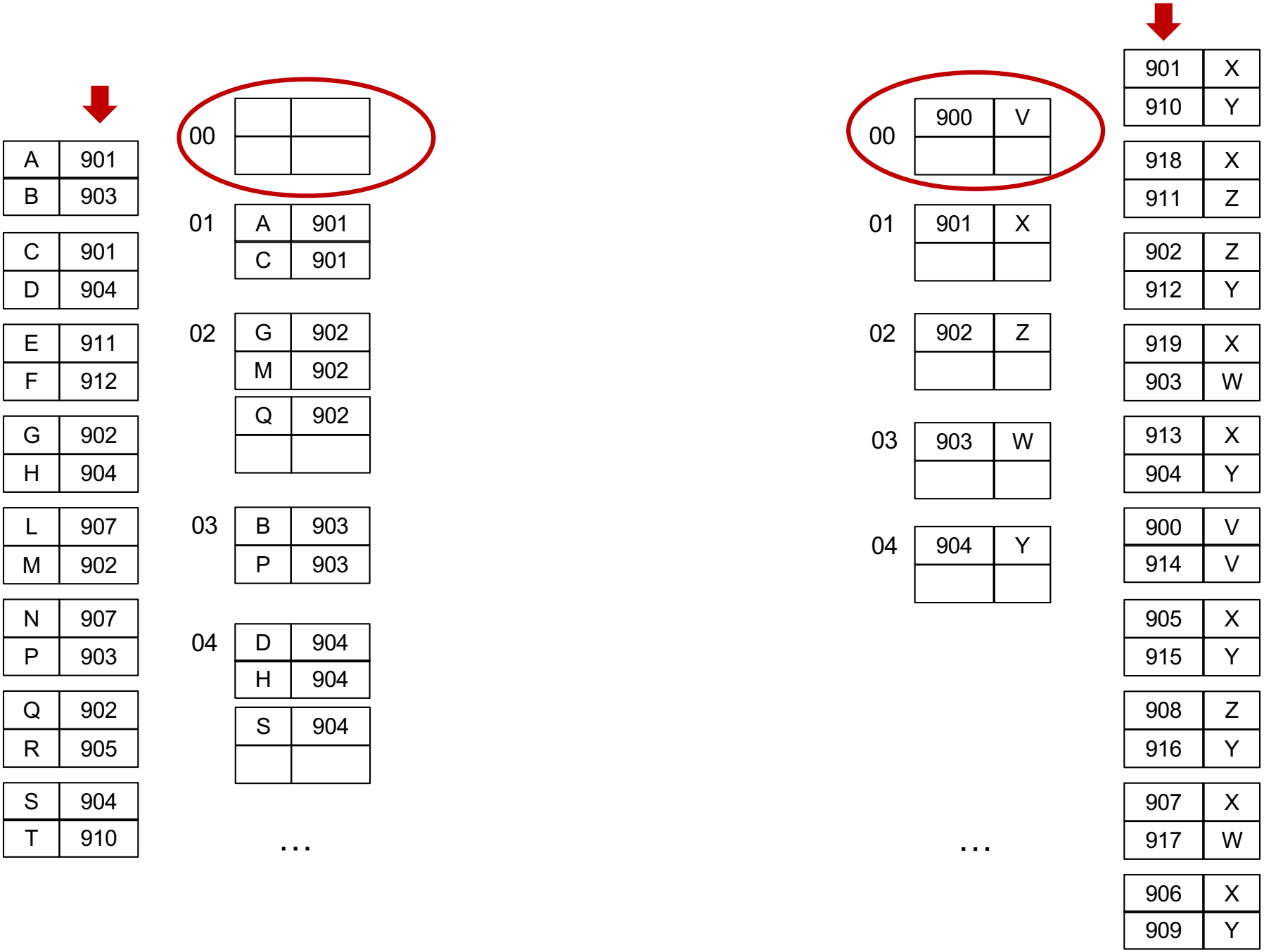

A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 902
N 907
P 903
Q 902
R 905
S 904
T 910
00
01
02
03
04
...
...
901 X
910 Y
918 X
911 Z
902 Z
912 Y
919 X
903 W
913 X
904 Y
900 V
914 V
905 X
915 Y
908 Z
916 Y
907 X
917 W
906 X
909 Y

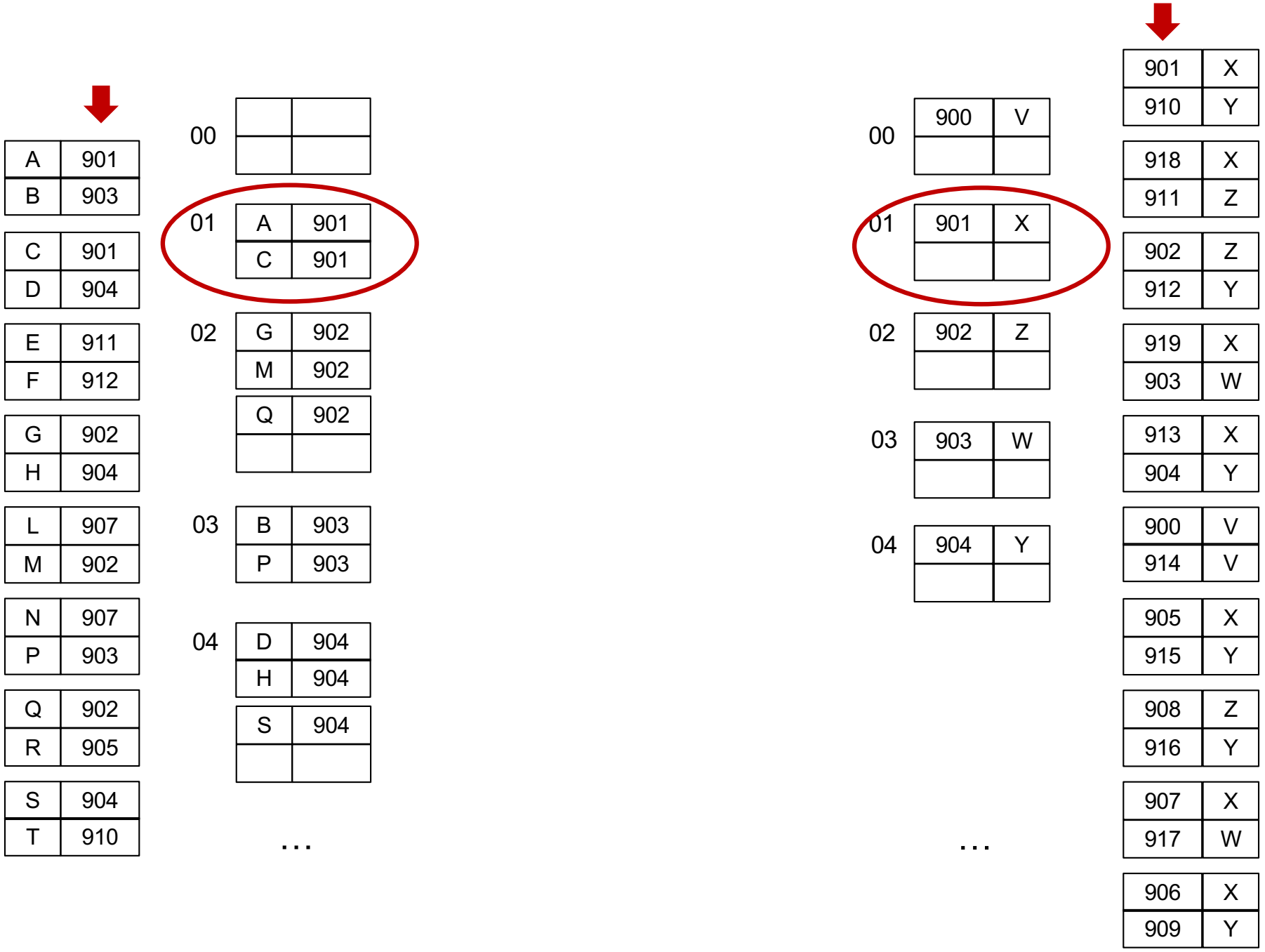

A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 902
N 907
P 903
Q 902
R 905
S 904
T 910
00
01 A 901
02
03
04
...
...
901 X
910 Y
918 X
911 Z
902 Z
912 Y
919 X
903 W
913 X
904 Y
900 V
914 V
905 X
915 Y
908 Z
916 Y
907 X
917 W
906 X
909 Y

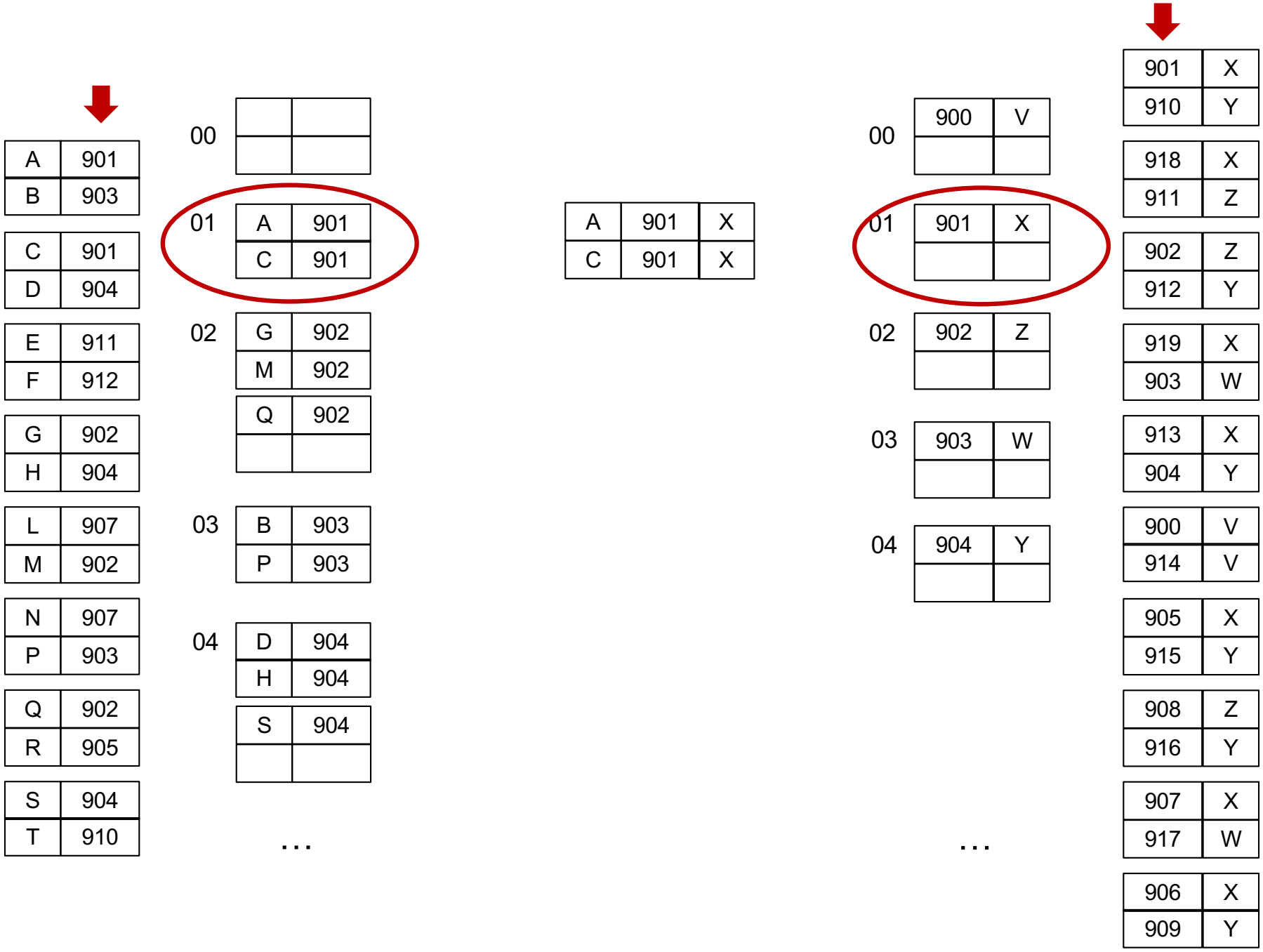

A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 902
N 907
P 903
Q 902
R 905
S 904
T 910
...
00
01 A 901
02
03 B 903
04
...
901 X
910 Y
918 X
911 Z
902 Z
912 Y
919 X
903 W
913 X
904 Y
900 V
914 V
905 X
915 Y
908 Z
916 Y
907 X
917 W
906 X
909 Y

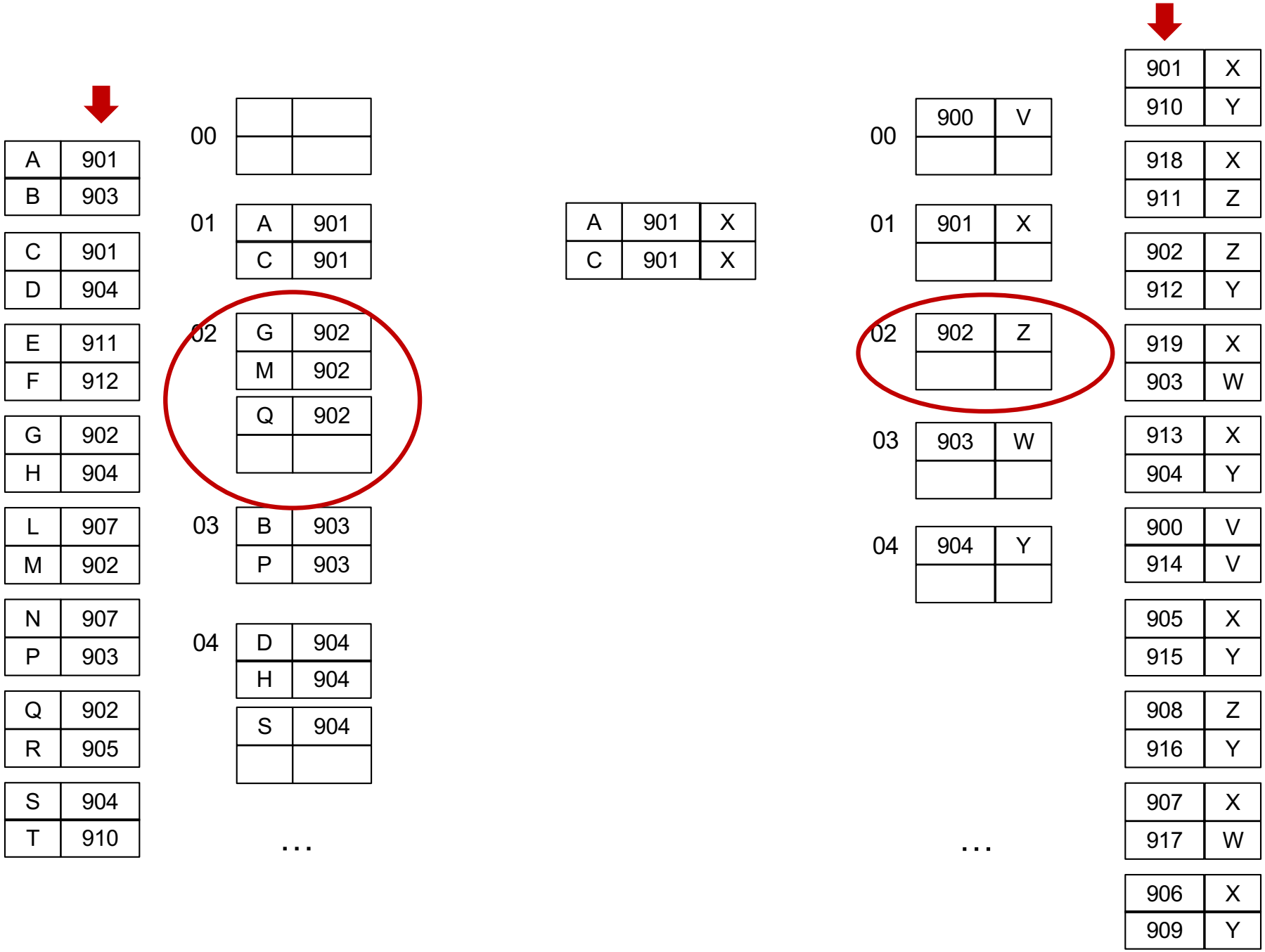

A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 902
N 907
P 903
Q 902
R 905
S 904
T 910
00
01 A 901 C 901
02 G 902 M 902 Q 902
03 B 903 P 903
04 D 904 H 904 S 904
...
...
901 X 910 Y
918 X 911 Z
902 Z 912 Y
919 X 903 W
913 X 904 Y
900 V 914 V
905 X 915 Y
908 Z 916 Y
907 X 917 W
906 X 909 Y

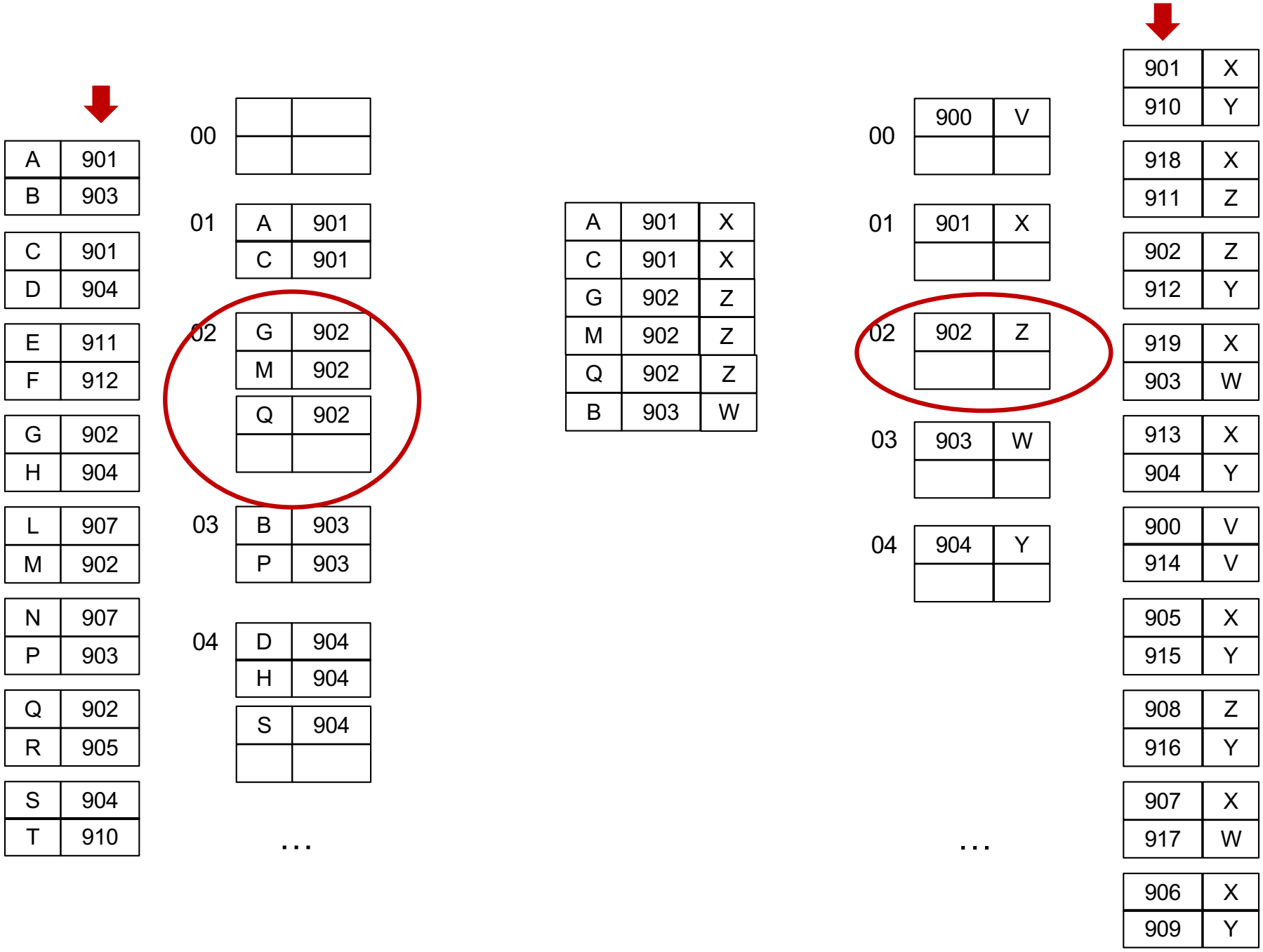

A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 902
N 907
P 903
Q 902
R 905
S 904
T 910
00
01 A 901 C 901
02 G 902 M 902 Q 902
03 B 903 P 903
04 D 904 H 904 S 904
...
00 900 V
01 901 X
02 902 Z
03 903 W
04 904 Y
...
901 X 910 Y
918 X 911 Z
902 Z 912 Y
919 X 903 W
913 X 904 Y
900 V 914 V
905 X 915 Y
908 Z 916 Y
907 X 917 W
906 X 909 Y

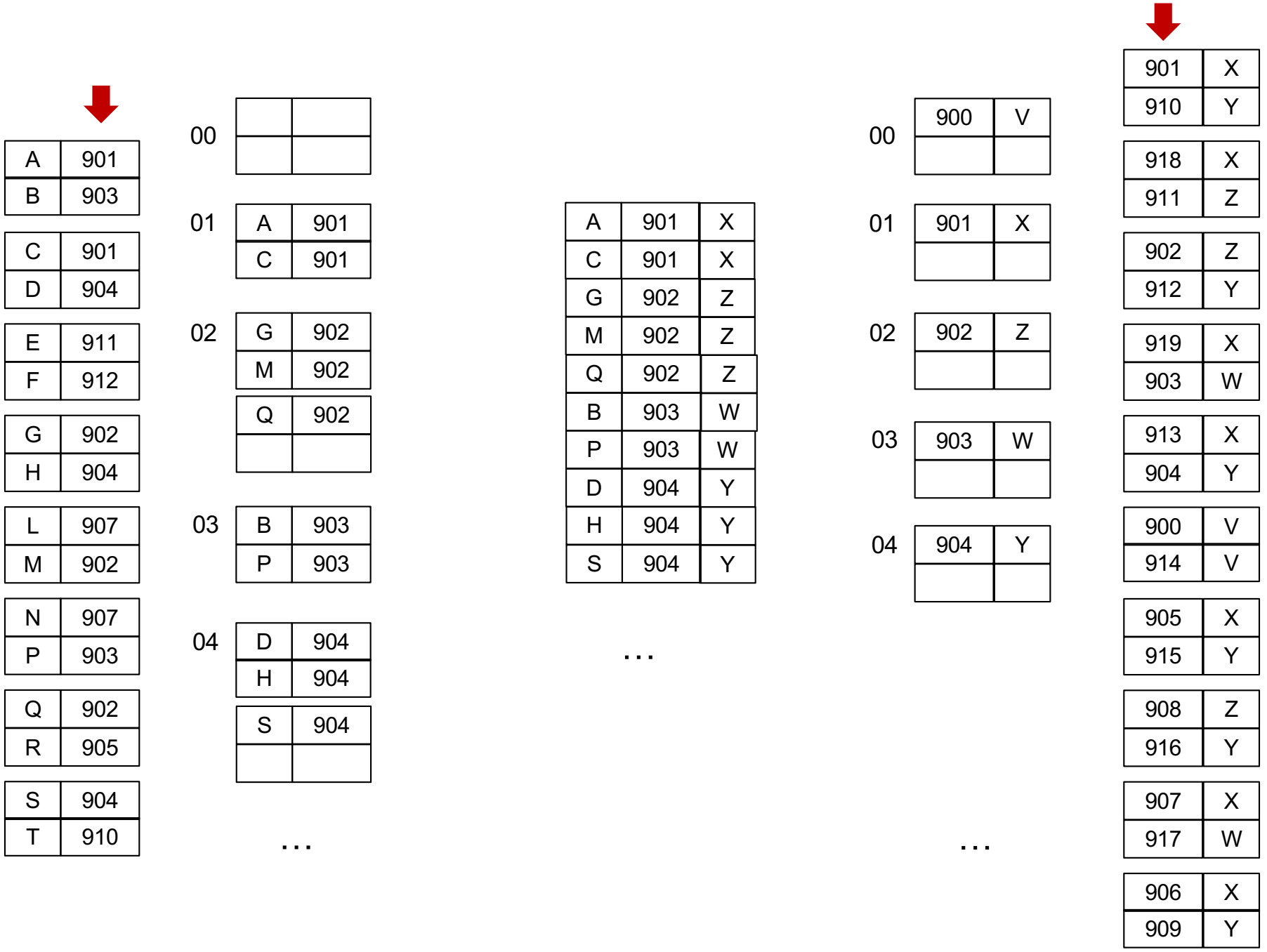

A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 902
N 907
P 903
Q 902
R 905
S 904
T 910
00
01 A 901 C 901
02 G 902 M 902 Q 902
03 B 903 P 903
04 D 904 H 904 S 904
...
00 900 V
01 901 X
02 902 Z
03 903 W
04 904 Y
...
901 X 910 Y
918 X 911 Z
902 Z 912 Y
919 X 903 W
913 X 904 Y
900 V 914 V
905 X 915 Y
908 Z 916 Y
907 X 917 W
906 X 909 Y


A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 902
N 907
P 903
Q 902
R 905
S 904
T 910
00
01 A 901 C 901
02 G 902 M 902 Q 902
03 B 903 P 903
04 D 904 H 904 S 904
...
00 900 V
01 901 X
02 902 Z
03 903 W
04 904 Y
...
901 X 910 Y
918 X 911 Z
902 Z 912 Y
919 X 903 W
913 X 904 Y
900 V 914 V
905 X 915 Y
908 Z 916 Y
907 X 917 W
906 X 909 Y


A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 902
N 907
P 903
Q 902
R 905
S 904
T 910
00
01 A 901 C 901
02 G 902 M 902 Q 902
03 B 903 P 903
04 D 904 H 904 S 904
...
A 901 X C 901 X
00 900 V
01 901 X
02 902 Z
03 903 W
04 904 Y
...
901 X 910 Y
918 X 911 Z
902 Z 912 Y
919 X 903 W
913 X 904 Y
900 V 914 V
905 X 915 Y
908 Z 916 Y
907 X 917 W
906 X 909 Y


A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 902
N 907
P 903
Q 902
R 905
S 904
T 910
00
01 A 901 C 901
02 G 902 M 902 Q 902
03 B 903 P 903
04 D 904 H 904 S 904
...
A 901 X C 901 X
00 900 V
01 901 X
02 902 Z
03 903 W
04 904 Y
...
901 X 910 Y
918 X 911 Z
902 Z 912 Y
919 X 903 W
913 X 904 Y
900 V 914 V
905 X 915 Y
908 Z 916 Y
907 X 917 W
906 X 909 Y


A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 902
N 907
P 903
Q 902
R 905
S 904
T 910
00
01 A 901 C 901
02 G 902 M 902 Q 902
03 B 903 P 903
04 D 904 H 904 S 904
...
A 901 X
C 901 X
G 902 Z
M 902 Z
Q 902 Z
B 903 W
00 900 V
01 901 X
02 902 Z
03 903 W
04 904 Y
...
901 X
910 Y
918 X
911 Z
902 Z
912 Y
919 X
903 W
913 X
904 Y
900 V
914 V
905 X
915 Y
908 Z
916 Y
907 X
917 W
906 X
909 Y


A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 902
N 907
P 903
Q 902
R 905
S 904
T 910
00
01 A 901 C 901
02 G 902 M 902 Q 902
03 B 903 P 903
04 D 904 H 904 S 904
...
A 901 X
C 901 X
G 902 Z
M 902 Z
Q 902 Z
B 903 W
P 903 W
D 904 Y
H 904 Y
S 904 Y
...
00 900 V
01 901 X
02 902 Z
03 903 W
04 904 Y
...
901 X
910 Y
918 X
911 Z
902 Z
912 Y
919 X
903 W
913 X
904 Y
900 V
914 V
905 X
915 Y
908 Z
916 Y
907 X
917 W
906 X
909 Y

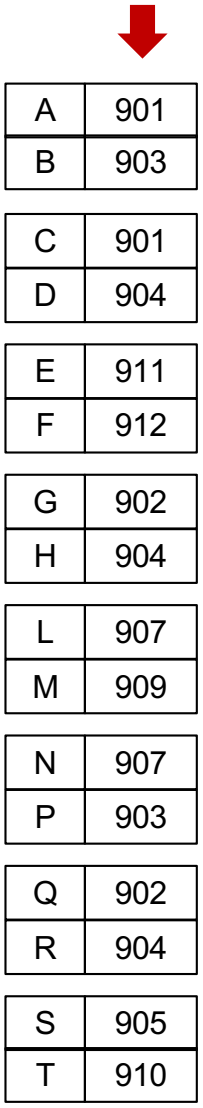
Hash-join, primi commenti

- File con N1 e N2 record e B1 e B2 blocchi
- Con un approccio semplice (migliorabile, come vedremo):
  - primo file viene letto sequenzialmente (costo B1);
  - i record, durante la lettura, vengono memorizzati secondo la funzione hash (costo N1)
  - secondo file viene letto sequenzialmente (costo B2);
  - i record, durante la lettura, vengono memorizzati secondo la funzione hash (costo N2)
  - poi si rilegge il tutto (B1 e B2) per "accoppiare"
    - B1 + N1 + B2 + N2 + B1 + B2
- Ma sfruttando i buffer si può fare molto meglio

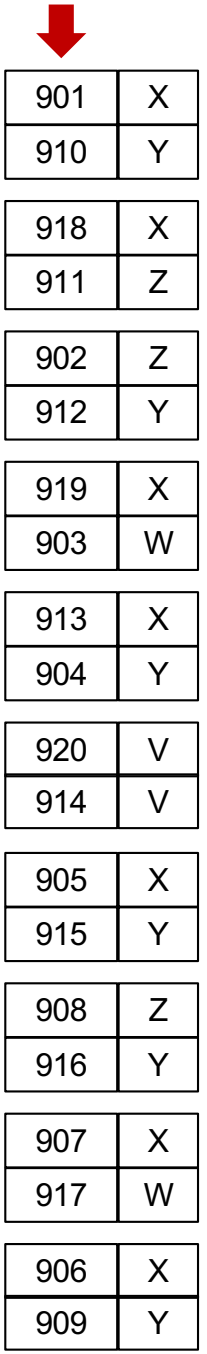
Hash-join con buffer

- Sfruttando i buffer, si può usare una funzione hash con numero di valori diversi paragonabile al numero di pagine di buffer P a disposizione

Hash join

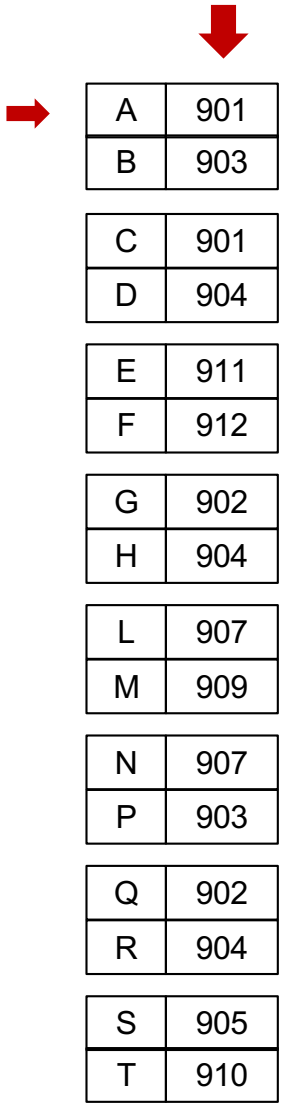


A	901
B	903
C	901
D	904
E	911
F	912
G	902
H	904
L	907
M	909
N	907
P	903
Q	902
R	904
S	905
T	910



901	X
910	Y
918	X
911	Z
902	Z
912	Y
919	X
903	W
913	X
904	Y
920	V
914	V
905	X
915	Y
908	Z
916	Y
907	X
917	W
906	X
909	Y

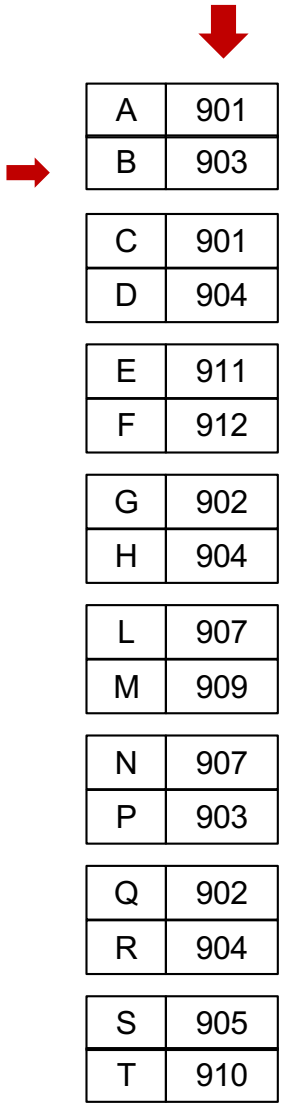
Primo file



Buffer

00		
01	A	901
02		

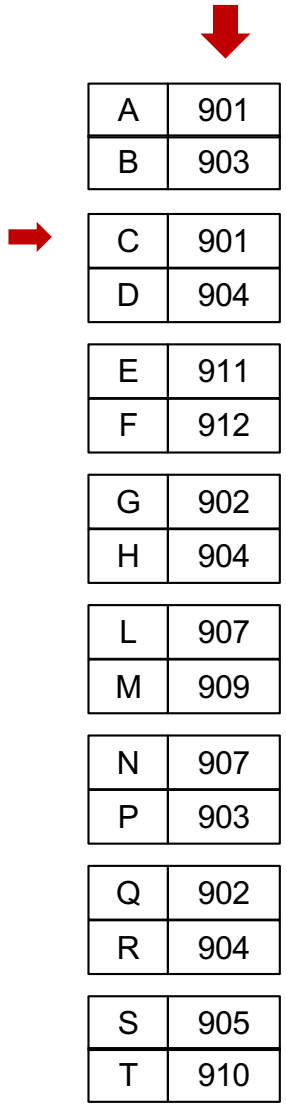
Primo file



Buffer

00	B	903
01	A	901
02		

Primo file



Buffer

00	B	903
01	A	901
	C	901
02		

Bucket

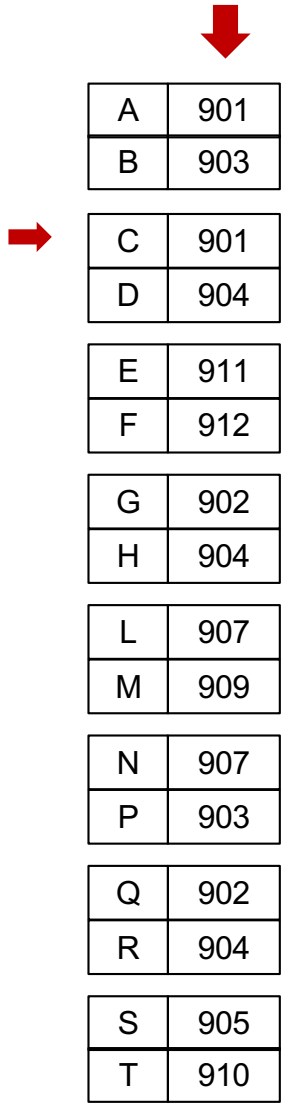
00

01

A	901
C	901

02

Primo file



Buffer

00	B	903
01	A	901
	C	901
02		

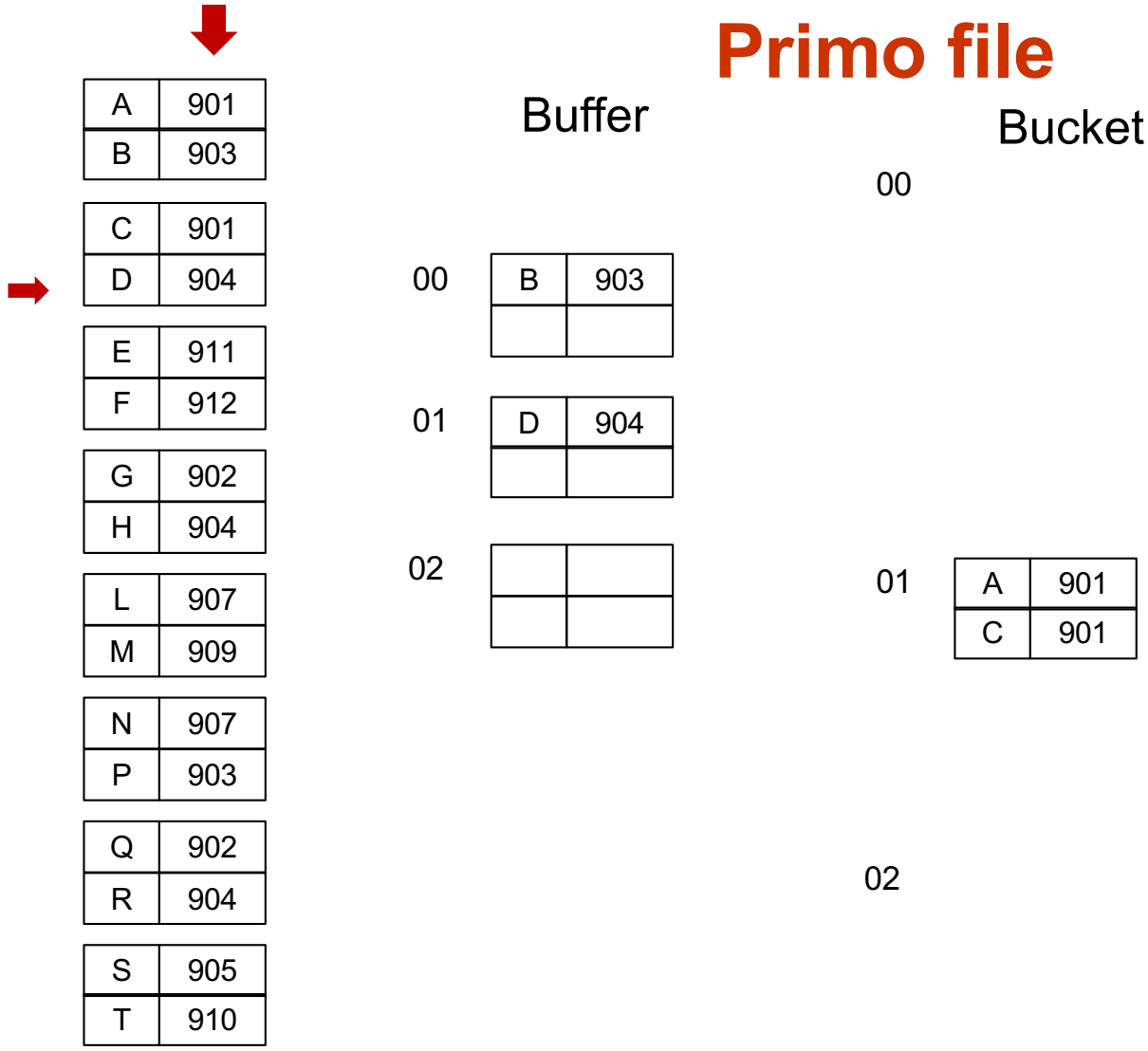
Bucket

00

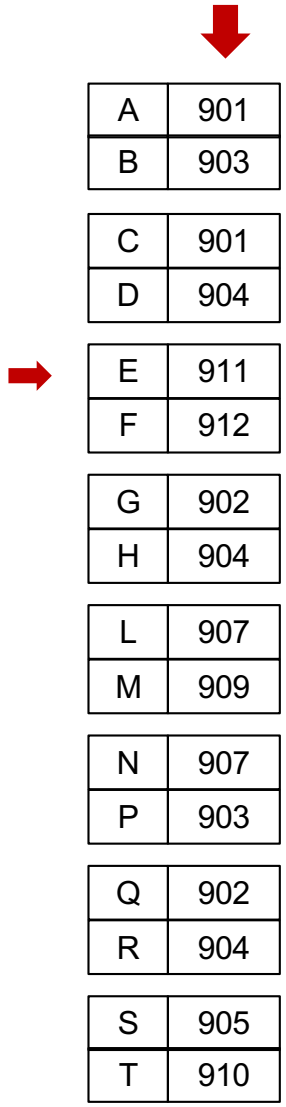
01

02

Primo file


A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 909
N 907
P 903
Q 902
R 904
S 905
T 910
Buffer
Bucket
00
B 903
01
D 904
02
00
01
A 901
C 901
02

Primo file



Buffer

00	B	903
01	D	904
02	E	911

Bucket

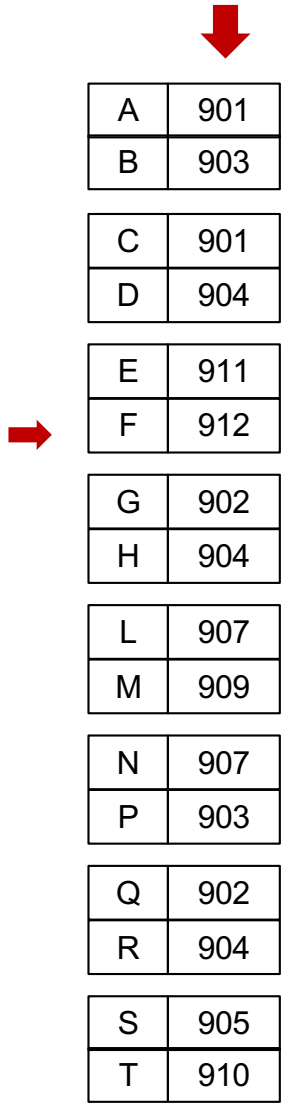
00

01

A	901
C	901

02

Primo file



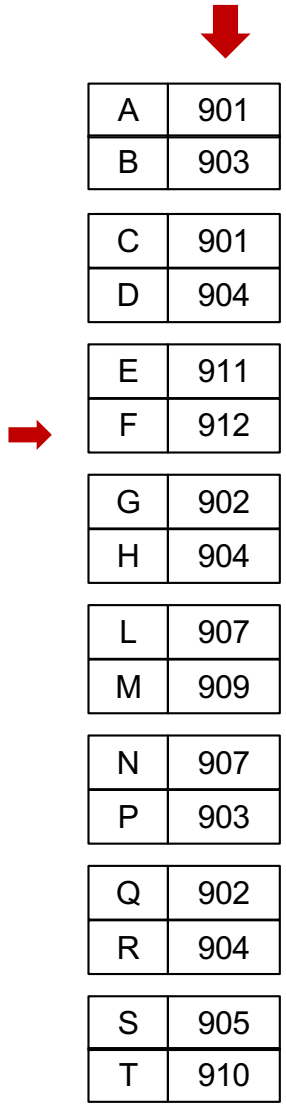
Buffer

00	B	903
	F	912
01	D	904
02	E	911

Bucket

00		
01	A	901
	C	901
02		

Primo file



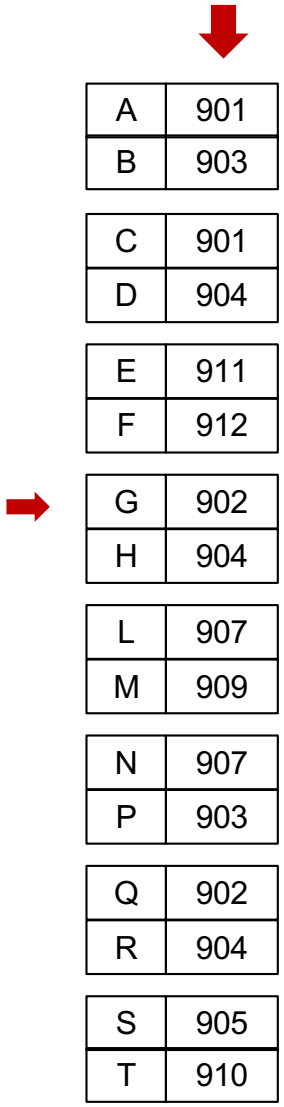
Buffer

00		
01	D	904
02	E	911

Bucket

00	B	903
	F	912
01	A	901
	C	901
02		

Primo file



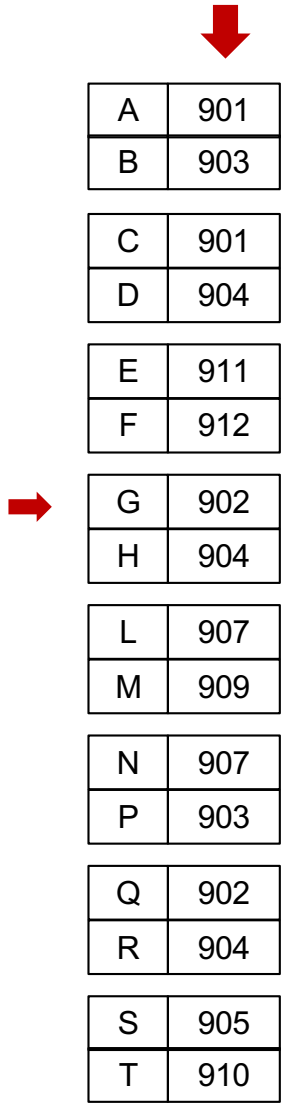
Buffer

00		
01	D	904
02	E	911
	G	902

Bucket

00	B	903
	F	912
01	A	901
	C	901
02		

Primo file



Buffer

00

01 D 904

02

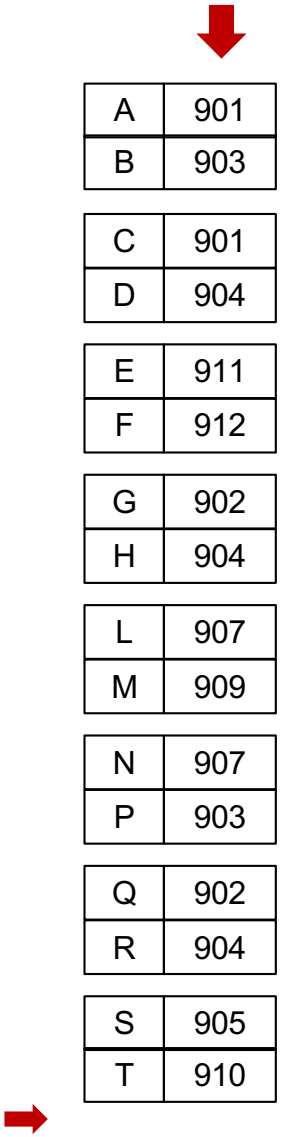
Bucket

00 B 903, F 912

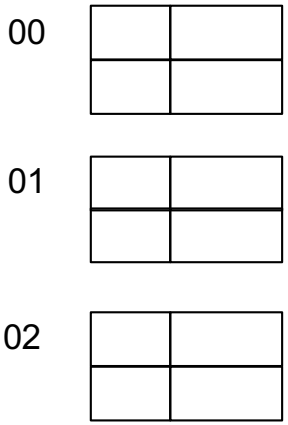
01 A 901, C 901

02 E 911, G 902

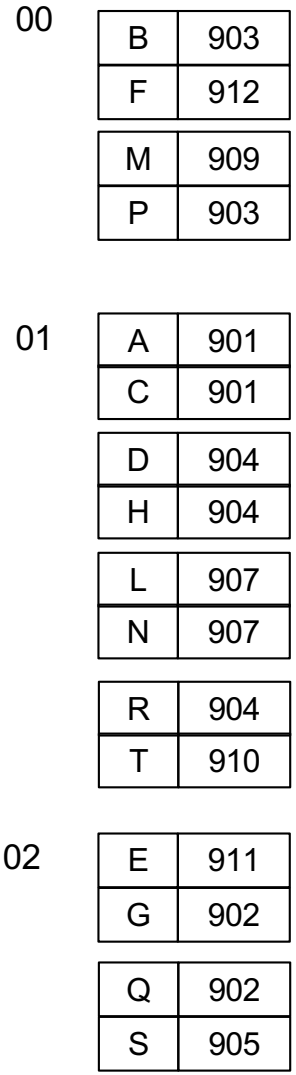
Primo file



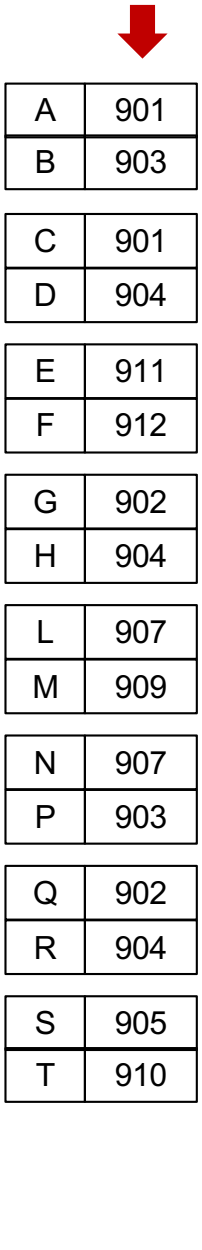
Buffer

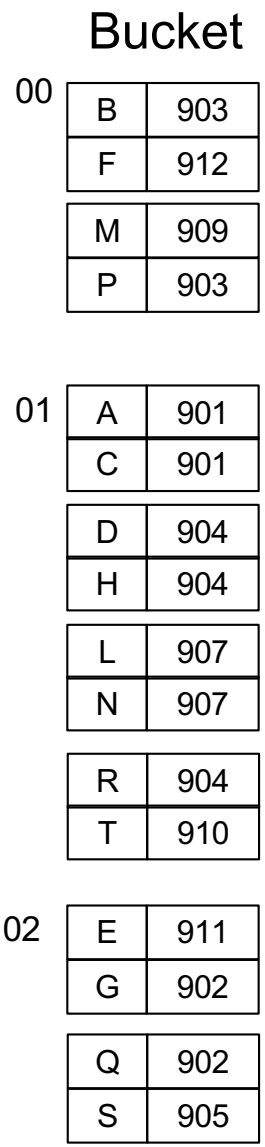


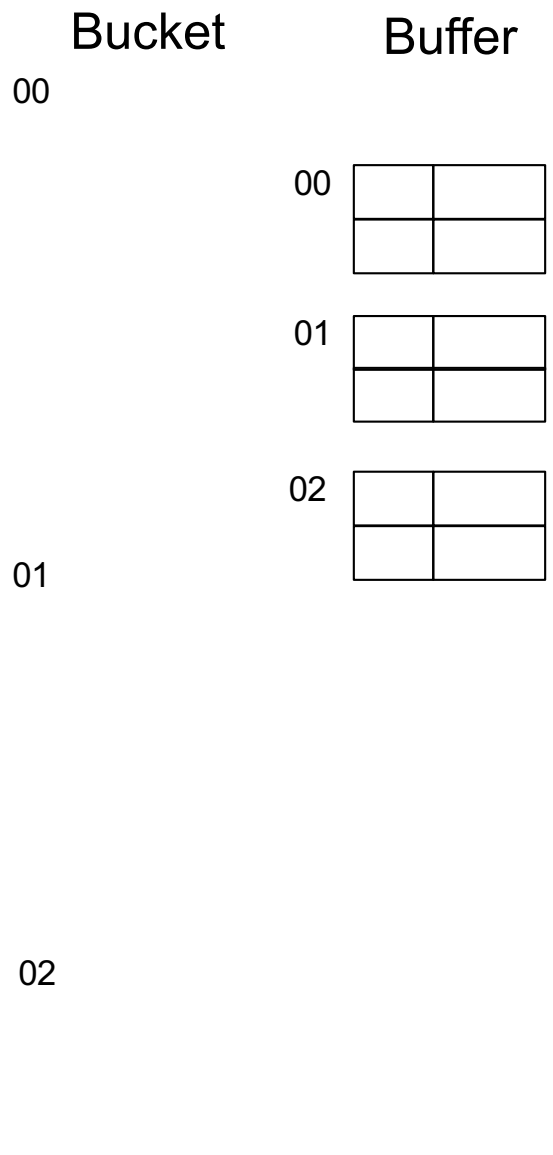
Bucket

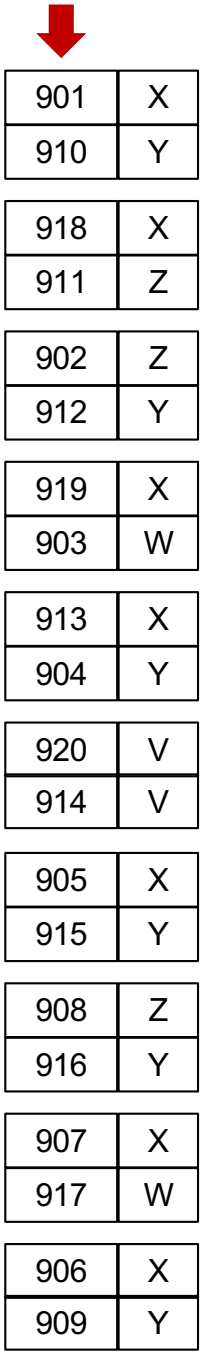


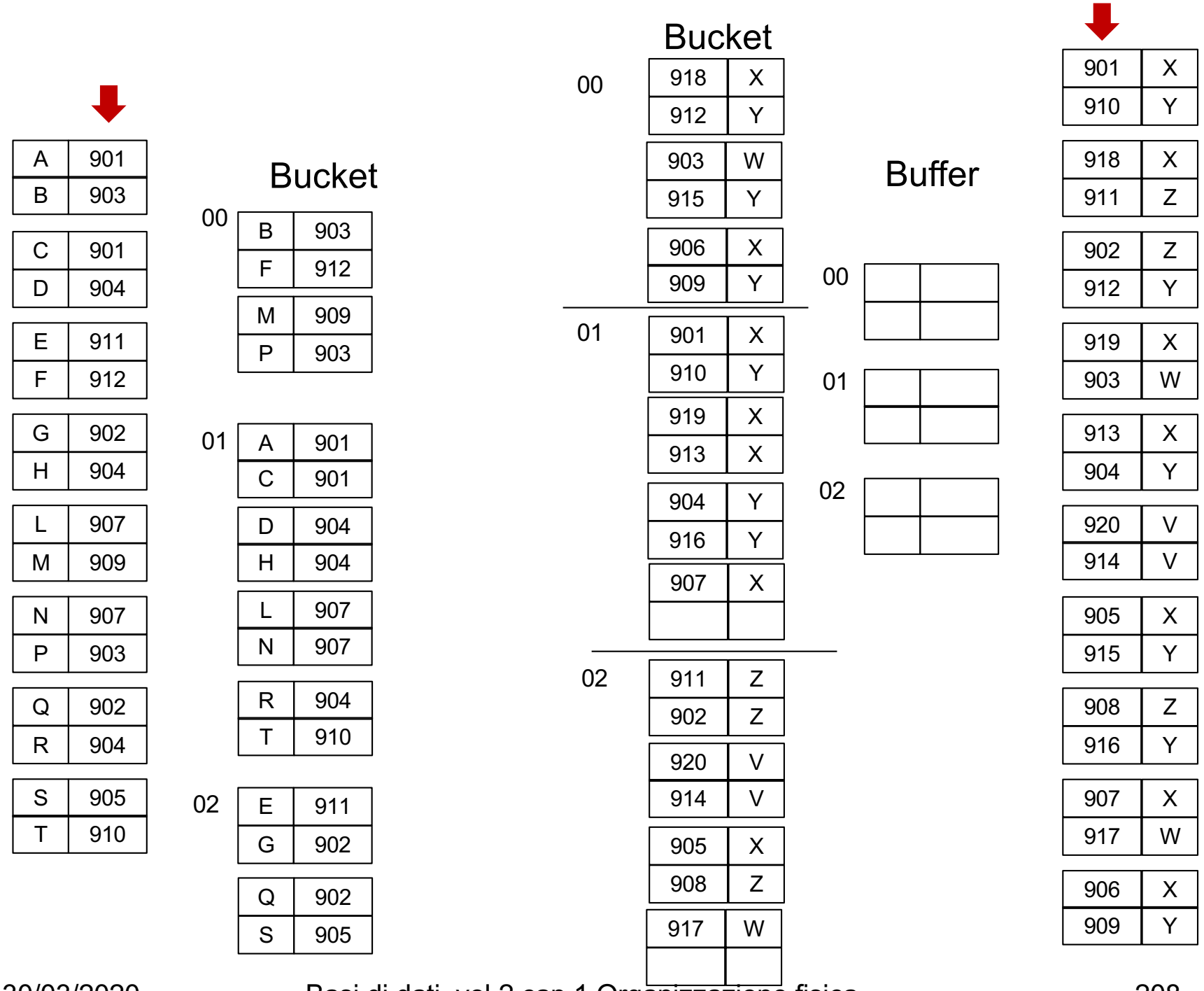
Secondo file

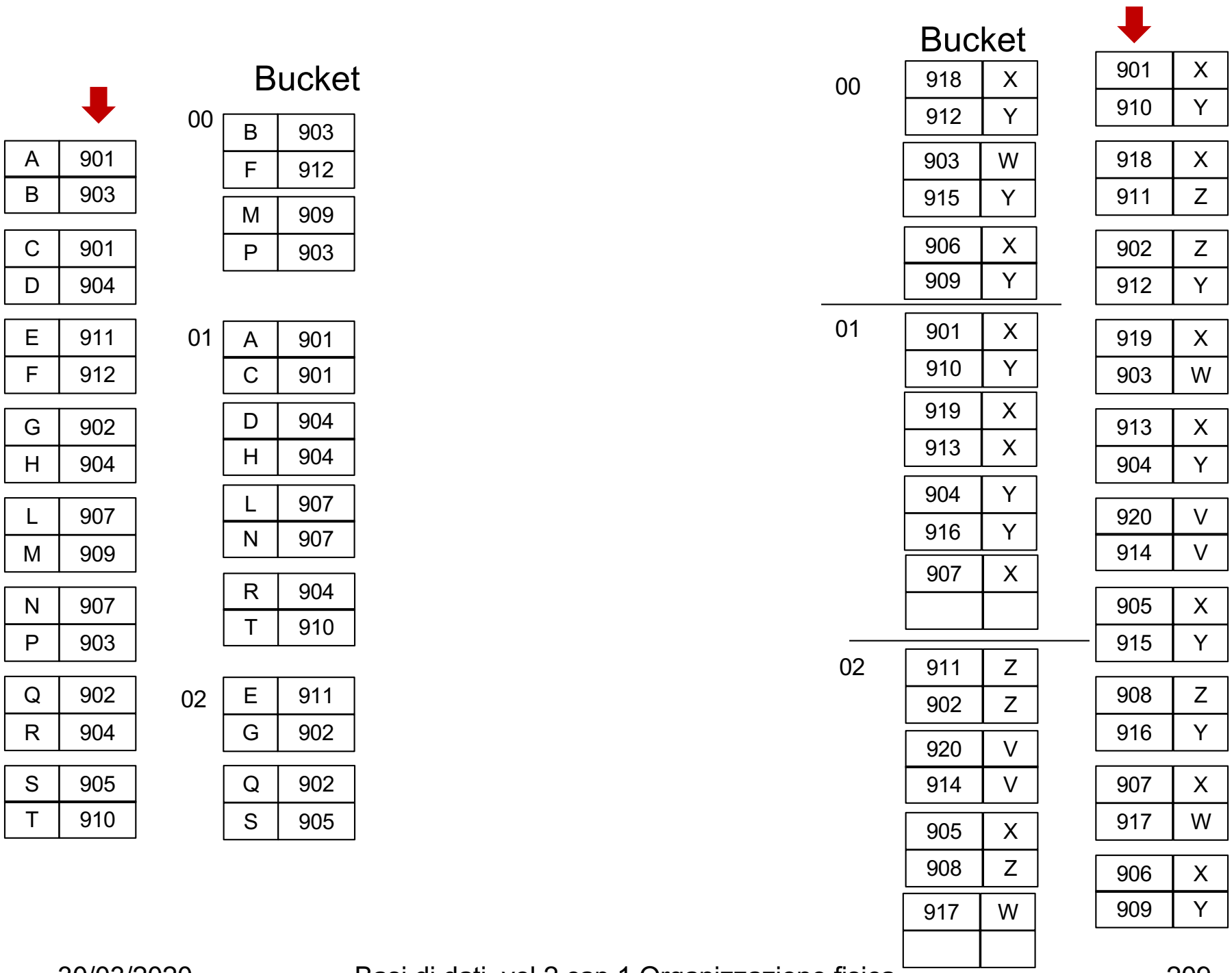


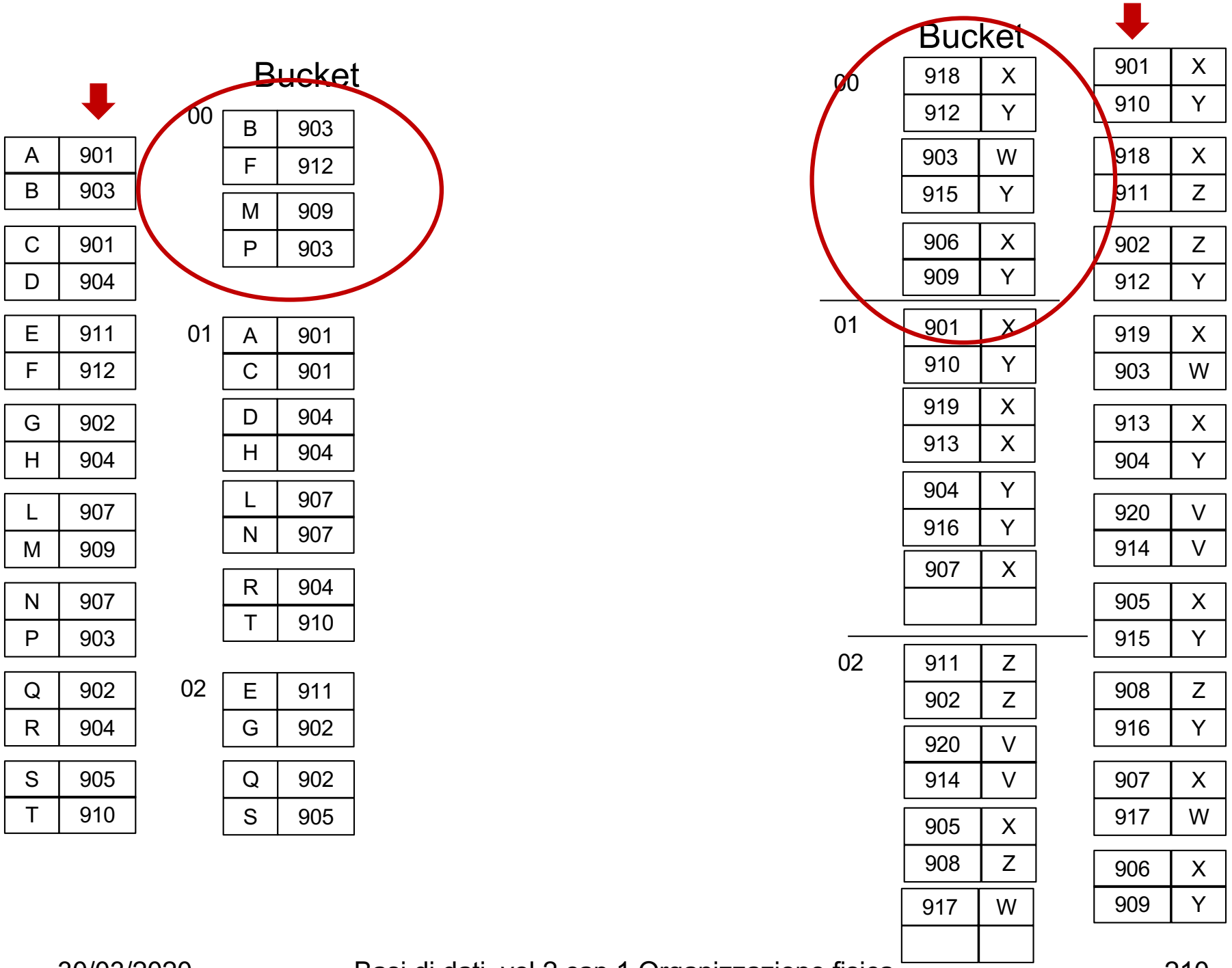


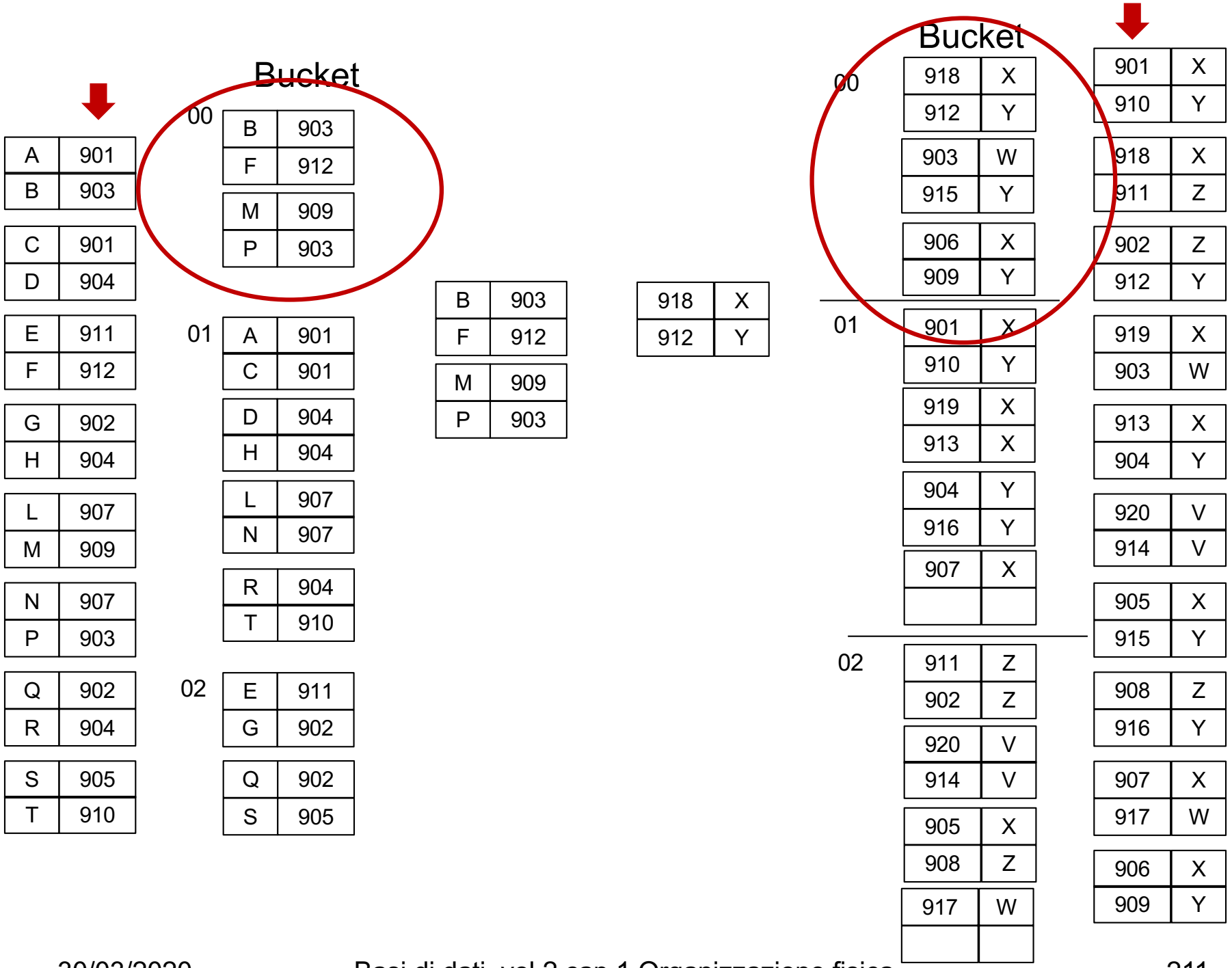


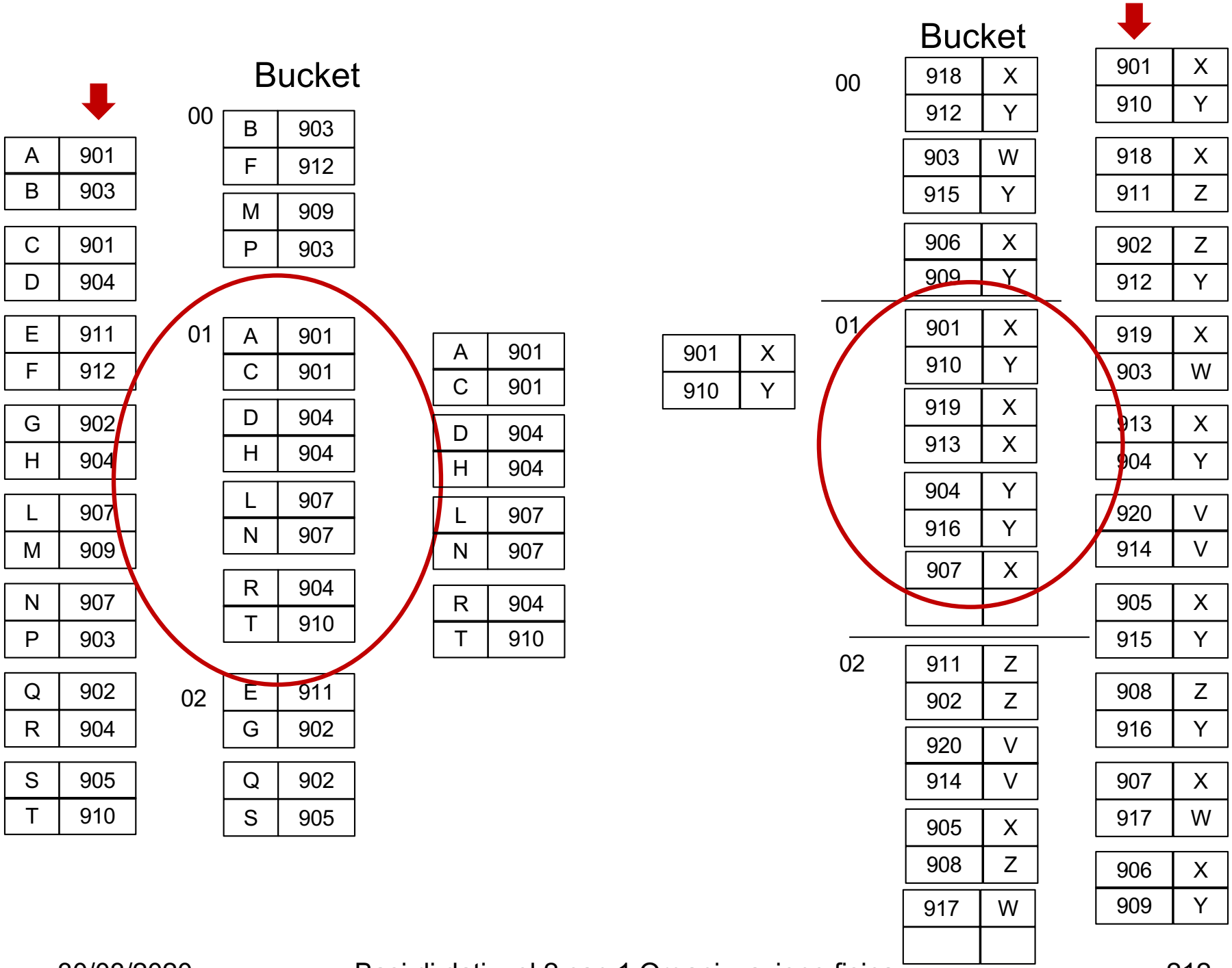



A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 909
N 907
P 903
Q 902
R 904
S 905
T 910
Bucket
00
B 903
F 912
M 909
P 903
01
A 901
C 901
D 904
H 904
L 907
N 907
R 904
T 910
02
E 911
G 902
Q 902
S 905
Bucket
00
918 X
912 Y
903 W
915 Y
906 X
909 Y
01
901 X
910 Y
919 X
913 X
904 Y
916 Y
907 X
02
911 Z
902 Z
920 V
914 V
905 X
908 Z
917 W
Buffer
00
01
02
901 X
910 Y
918 X
911 Z
902 Z
912 Y
919 X
903 W
913 X
904 Y
920 V
914 V
905 X
915 Y
908 Z
916 Y
907 X
917 W
906 X
909 Y


Bucket
00
01
02
A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 909
N 907
P 903
Q 902
R 904
S 905
T 910
B 903
F 912
M 909
P 903
A 901
C 901
D 904
H 904
L 907
N 907
R 904
T 910
E 911
G 902
Q 902
S 905
Bucket
00
01
02
918 X
912 Y
903 W
915 Y
906 X
909 Y
901 X
910 Y
919 X
913 X
904 Y
916 Y
907 X
911 Z
902 Z
920 V
914 V
905 X
908 Z
917 W
901 X
910 Y
918 X
911 Z
902 Z
912 Y
919 X
903 W
913 X
904 Y
920 V
914 V
905 X
915 Y
908 Z
916 Y
907 X
917 W
906 X
909 Y


Bucket
00
01
02
A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 909
N 907
P 903
Q 902
R 904
S 905
T 910
B 903
F 912
M 909
P 903
A 901
C 901
D 904
H 904
L 907
N 907
R 904
T 910
E 911
G 902
Q 902
S 905
Bucket
00
01
02
918 X
912 Y
903 W
915 Y
906 X
909 Y
901 X
910 Y
919 X
913 X
904 Y
916 Y
907 X
911 Z
902 Z
920 V
914 V
905 X
908 Z
917 W
901 X
910 Y
918 X
911 Z
902 Z
912 Y
919 X
903 W
913 X
904 Y
920 V
914 V
905 X
915 Y
908 Z
916 Y
907 X
917 W
906 X
909 Y


Bucket
00
01
02
A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 909
N 907
P 903
Q 902
R 904
S 905
T 910
B 903
F 912
M 909
P 903
A 901
C 901
D 904
H 904
L 907
N 907
R 904
T 910
E 911
G 902
Q 902
S 905
B 903
F 912
M 909
P 903
918 X
912 Y
Bucket
00
01
02
918 X
912 Y
903 W
915 Y
906 X
909 Y
901 X
910 Y
919 X
913 X
904 Y
916 Y
907 X
911 Z
902 Z
920 V
914 V
905 X
908 Z
917 W
901 X
910 Y
918 X
911 Z
902 Z
912 Y
919 X
903 W
913 X
904 Y
920 V
914 V
905 X
915 Y
908 Z
916 Y
907 X
917 W
906 X
909 Y


Bucket
00
01
02
A 901
B 903
C 901
D 904
E 911
F 912
G 902
H 904
L 907
M 909
N 907
P 903
Q 902
R 904
S 905
T 910
B 903
F 912
M 909
P 903
A 901
C 901
D 904
H 904
L 907
N 907
R 904
T 910
E 911
G 902
Q 902
S 905
A 901
C 901
D 904
H 904
L 907
N 907
R 904
T 910
901 X
910 Y
Bucket
00
01
02
918 X
912 Y
903 W
915 Y
906 X
909 Y
901 X
910 Y
919 X
913 X
904 Y
916 Y
907 X
911 Z
902 Z
920 V
914 V
905 X
908 Z
917 W
901 X
910 Y
918 X
911 Z
902 Z
912 Y
919 X
903 W
913 X
904 Y
920 V
914 V
905 X
915 Y
908 Z
916 Y
907 X
917 W
906 X
909 Y

Hash-join con buffer, costi

- Sfruttando i buffer, si può usare una funzione hash con numero di valori diversi paragonabile al numero di pagine di buffer P a disposizione e **così si può spesso eseguire l'algoritmo in due passate di lettura più una scrittura**:
  - Ciascun file viene letto e riorganizzato in P liste di blocchi (costo Bi per la lettura e Bi per la riscrittura, numero di blocchi e non di record perché si scrivono blocchi completi) poi, per ciascun valore della funzione hash si confrontano le liste omologhe (costo B1 + B2 per la lettura).
  - L'algoritmo non richiede altri accessi se le singole partizioni di uno dei file entrano in memoria, cioè se min(B1,B2)/P < P cioè se P^2 > min(B1,B2). In tal caso il costo è 3(B1+B2) + il numero di blocchi del risultato (se va scritto)
  - Nota i calcoli sono approssimati, perché la funzione hash ha una distribuzione