Prefix String Map of Old PrefixSet

4	0001 1111 2222 FFFF
7	0123456 FAAAAAA
9	987654321 F876543210
11	BBBBBBBBBBB

Prefix String Map from update

4	3333 4567 890A BCDE
5	A0000 B1111 C2222
7	DDDDFFF
6	55665566 99999999
11	00112233445

Removal Indices from update

1	5	7

The goal of partial update algorithm is to merge two prefix string map into one

PrefixStringMap is a hash table
Key: Prefix size
Value : Lexicographic-sorted strings of all prefixes with the same prefix size.

Prefix String Map of Old PrefixSet

4	0001 1111 2222 FFFF
7	0123456 FAAAAAA
9	987654321 F876543210
11	BBBBBBBBBBB

Step1. Pick smallest one from old map by comparing start elements of each prefix string(**red color**)

=> get 0001

Prefix String Map from update

4	3333 4567 890A BCDE
5	A0000 B1111 C2222
7	DDDDFFF
6	55665566 99999999
11	00112233445

Step2. Pick smallest one from add map by comparing start elements of each prefix string(**red color**)

=> get 00112233445

Removal Indices from update

1	5	7

Step3. compare 0001 & 00112233445

=> get 0001

Since 0001 is from old map,increase numOldPrefixPicked to **0**

Prefix String Map of New PrefixSet

Append 0001 to new map

Prefix String Map of Old PrefixSet

4	~~0001~~ 1111 2222 FFFF
7	0123456 FAAAAAA
9	987654321 F876543210
11	BBBBBBBBBBB

Step1. Pick smallest one from old map again by comparing start elements of each prefix string(**red color**), start element now point to 1111 instead of 0001

=> get 0123456

Prefix String Map from update

4	3333 4567 890A BCDE
5	A0000 B1111 C2222
7	DDDDFFF
6	55665566 99999999
11	00112233445

Step2. 00112233445 is picked last round, just use it

=> get 00112233445

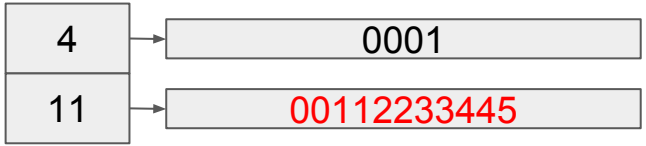
Removal Indices from update

1	5	7

Step3. compare 0123456 & 00112233445

=> get 00112233445

Prefix String Map of Merged PrefixSet



Append 00112233445 to new map

Prefix String Map of Old PrefixSet

4	~~0001~~ 1111 2222 FFFF
7	0123456 FAAAAAA
9	987654321 F876543210
11	BBBBBBBBBBB

Step1. 0123456 is picked last round, just use it

=> get 0123456

Prefix String Map from update

4	3333 4567 890A BCDE
5	A0000 B1111 C2222
7	DDDDFFF
6	55665566 99999999
11	~~00112233445~~

Step2. Pick smallest one from add map by comparing start elements of each prefix string(**red color**)

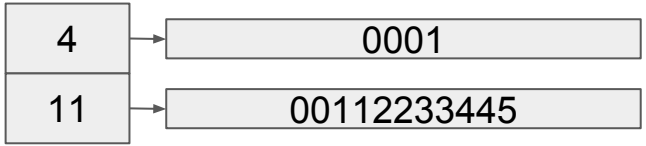
=> get 3333

Removal Indices from update

1	5	7

Step3. compare 0123456 & 3333
=> get 0123456
Since 0123456 is from old map,increase numOldPrefixPicked to **1**, **1 is equal to the first element of removal indices array, so just don't merge this to new map**

Prefix String Map of Merged PrefixSet



Remain the same because 0123456 is not added to map

Prefix String Map of Old PrefixSet

4	~~0001~~ 1111 2222 FFFF
7	~~0123456~~ FAAAAAA
9	987654321 F876543210
11	BBBBBBBBBBB

Prefix String Map from update

4	3333 4567 890A BCDE
5	A0000 B1111 C2222
7	DDDDFFF
6	55665566 99999999
11	~~00112233445~~

Removal Indices from update

5	7

Just keep doing the same things until old map or add map is empty, if one of the map is empty, then we can merge the remaining map to new map directly.

But there is a special case when add map is emprty(see next slide)

Prefix String Map of Old PrefixSet

4	~~0001~~ ~~1111~~ ~~2222~~ FFFF
7	~~0123456~~ FAAAAAA
9	~~987654321~~ F876543210
11	~~BBBBBBBBBBB~~

Prefix String Map from update

4	~~3333~~ ~~4567~~ ~~890A~~ ~~BCDE~~
5	~~A0000~~ ~~B1111~~ ~~C2222~~
7	~~DDDDFFF~~
6	~~55665566~~ ~~99999999~~
11	~~00112233445~~

Removal Indices from update

7

Add map is empty now.
There are still three prefixes in old map.

Since removal indices is not yet empty, point to 8 now.

We will still need

1. Compare FFFF & FAAAAAA & F876543210

=> get F876543210 (numOldPrefixPicked is 6 now)

2. Compare FFFF & FAAAAAA

=> get FAAAAAA (numOldPrefixPicked is 7 now)

Do not merge FAAAAA

3. Merge FFFF to new Map