Constraint :: All the Operations average time complexity should be O(1)

Design Shuffling PlayList.

public boolean addSong(int songId)

returns true if song added else false.

public boolean removeSong(int songId)

returns true if song removed else false.

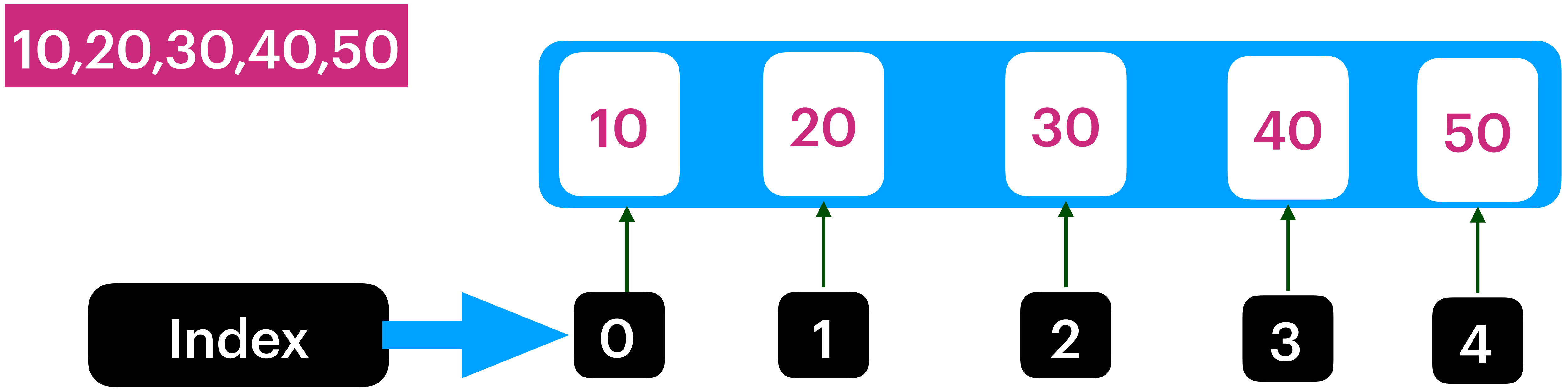
public boolean getRandom()

Should return Random song

Example :

```
addSong(7); => true
addSong(15); => true
addSong(3); => true
getRandom(); => can return 7 or 15 or 3
addSong(15) => false
removeSong(15); => true
getRandom(); => can return 7 or 3
removeSong(15); => false
```

ArrayList : 10,20,30,40,50

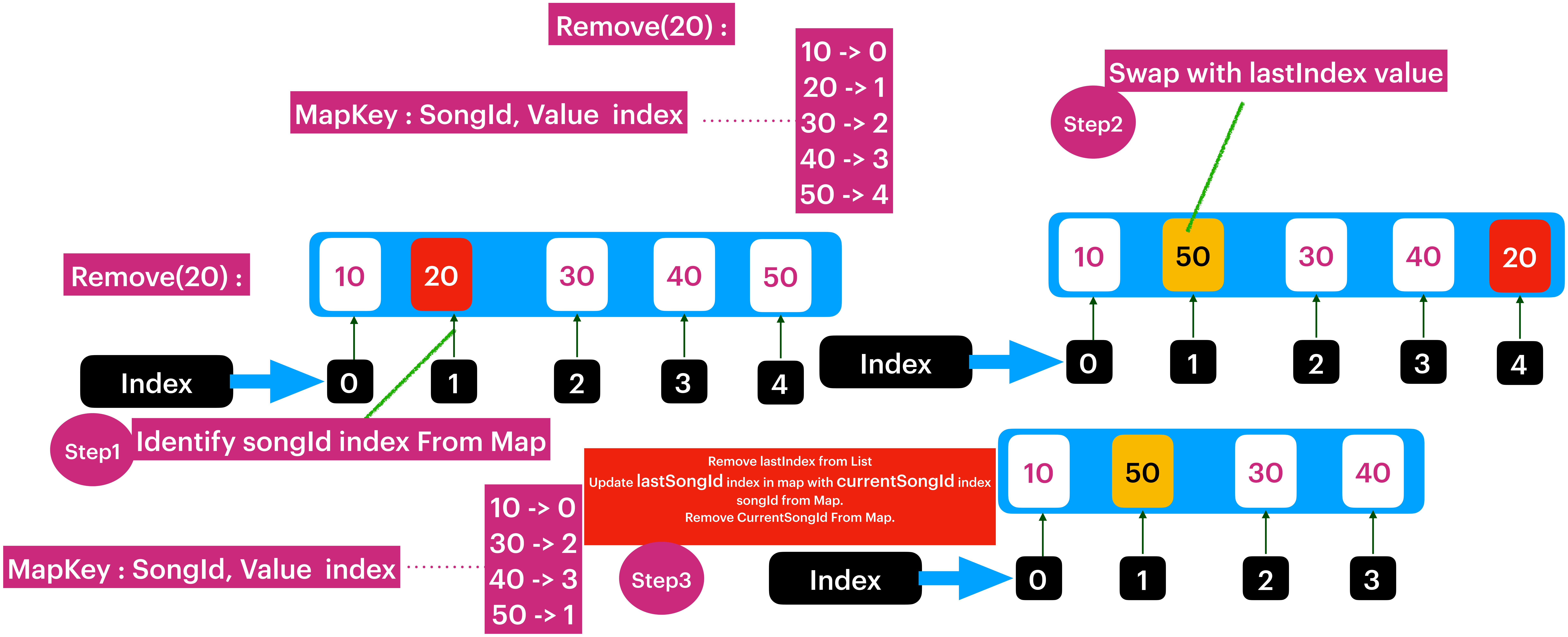


Add : O(1)

getRandom : O(1) ?

MapKey : SongId, Value index

10 -> 0
20 -> 1
30 -> 2
40 -> 3
50 -> 4



Remove(20) :

MapKey : SongId, Value index

10 -> 0
20 -> 1
30 -> 2
40 -> 3
50 -> 4

Remove(20) :

Step1 Identify songId index From Map

Step2 Swap with lastIndex value

Remove lastIndex from List
Update lastSongId index in map with currentSongId index songId from Map.
Remove CurrentSongId From Map.

Step3

MapKey : SongId, Value index

10 -> 0
30 -> 2
40 -> 3
50 -> 1