

## Selection Sort

One way is to go through the list and find the most-played artist. Add that artist to a new list.

~🎵~	PLAY COUNT		🎵 SORTED 🎵	PLAY COUNT
RADIOHEAD	156	→	RADIOHEAD	156
KISHORE KUMAR	141			
THE BLACK KEYS	35			
NEUTRAL MILK HOTEL	94			
BECK	88			
THE STROKES	61			
WILCO	111			

1. RADIOHEAD IS THE MOST PLAYED ARTIST...

2. ADD IT TO A NEW LIST

Do it again to find the next-most-played artist.

~🎵~	PLAY COUNT		🎵 SORTED 🎵	PLAY COUNT
		→	RADIOHEAD	156
KISHORE KUMAR	141		KISHORE KUMAR	141
THE BLACK KEYS	35			
NEUTRAL MILK HOTEL	94			
BECK	88			
THE STROKES	61			
WILCO	111			

1. KISHORE KUMAR IS THE NEXT MOST-PLAYED ARTIST

2. SO IT IS THE NEXT ARTIST ADDED TO THE NEW LIST

Keep doing this, and you'll end up with a sorted list.

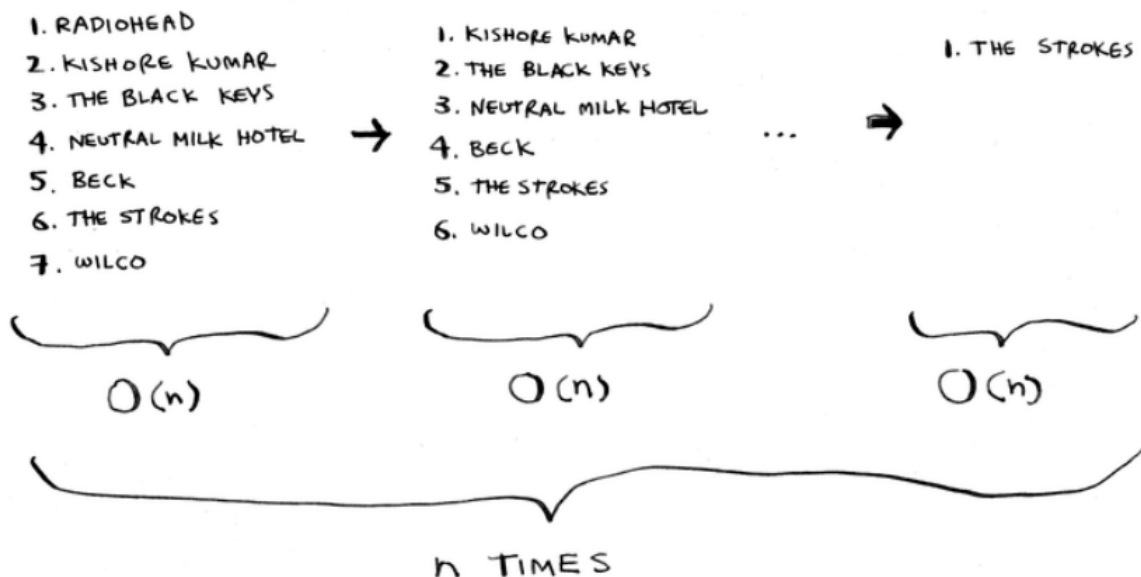
~🎵~	PLAY COUNT
RADIOHEAD	156
KISHORE KUMAR	141
WILCO	111
NEUTRAL MILK HOTEL	94
BECK	88
THE STROKES	61
THE BLACK KEYS	35

- Selection sort takes  $O(n^2)$  time.
- It actually takes  $n*(n+1)/2$  time ( $n^2+n/2$ ), however since we don't care about constants, it is  $n^2$

Let's put on our computer science hats and see how long this will take to run. Remember that  $O(n)$  time means you touch every element in a list once. For example, running simple search over the list of artists means looking at each artist once.

1. RADIOHEAD
  2. KISHORE KUMAR
  3. THE BLACK KEYS
  4. NEUTRAL MILK HOTEL
  5. BECK
  6. THE STROKES
  7. WILCO
- }  $n$  ITEMS

To find the artist with the highest play count, you have to check each item in the list. This takes  $O(n)$  time, as you just saw. So you have an operation that takes  $O(n)$  time, and you have to do that  $n$  times:



This takes  $O(n \times n)$  time or  $O(n^2)$  time.

- Selection sort is a neat algorithm, but it's not very fast. Quicksort is a faster sorting algorithm that only takes  $O(n \log n)$  time. It's coming up in the next chapter!