SORTOWANIE KUBEŁKOWE- NAJGORSZA ZŁOŻONOŚĆ: O(n^2)

Najgorszy przypadek to O (n^2).

Schemat ogólny liczenia złożoności bucketsort:

1. Dodaj element do wiadra. Używanie listy połączonej to O (1)
2. Przechodząc przez listę i umieszczając elementy we właściwym wiadrze = O (n)
3. Łączenie segmentów = O (k)
4. O (1) * O (n) + O (k) = O (n + k)

Otrzymujemy O(n+k). W założeniu sortowania kubełkowego rozłożenie liczb jest równomierne. Jednak jeśli algorytm zdecyduje, że każdy element należy do tego samego kubełka to w takim przypadku lista połączona w tym segmencie musi być wykonywana za każdym razem, gdy dodawany jest element. To zajmuje 1 krok, potem 2, potem 3, 4, 5, aż do *n*. Zatem czas jest sumą wszystkich liczb od 1 do *n,* która jest (n ^ 2 + n) / 2, czyli O (n ^ 2).
Oczywiście jest to „najgorszy przypadek" (wszystkie elementy w jednym wiadrze) - algorytm do obliczania, które wiadro umieścić element, jest zazwyczaj zaprojektowany tak, aby uniknąć tego zachowania.

W sortowaniu kubełkowym liczba segmentów n musi być równa długości sortowanej tablicy, a tablica wejściowa musi być równomiernie rozłożona w zakresie możliwych wartości segmentów. Jeśli te wymagania nie zostaną spełnione, wydajność sortowania będzie zdominowana przez czas działania nextSort, który zwykle jest sortowaniem wstawiania O (n ^ {2}).