Php’d e random sayıları while ile oluşturarak, insertion sıralama yapar, çalışma zamanını ve hafıza da kapladığı yeri hesaplar.

<?php

$sure\_baslangici = microtime(true);

$sayi=rand(1,1000);

$sayac=1;

$sayilar[]=$sayi;

while($sayac<1000){

$sayi=rand(1,1000);

$varmi=false; // her dizi elemanı üretilen sayı ile karşılaştırılacak

$indis=0;

while(@$sayilar[$indis]){

if ($sayilar[$indis]==$sayi) //eğer üretilen sayı dizide varsa

$varmi=true; // dizide var olarak işaretle

$indis++;

}

if ($varmi==false){ // eğer üretilen sayı dizide yoksa

$sayilar[]=$sayi; // diziye ekle

$sayac++;

}

}

$indis=0;

while(@$sayilar[$indis]){

echo $sayilar[$indis]," ";

$indis++;

}

function insertion\_Sort($sayilar)

{

for($i=0;$i<count($sayilar);$i++){

$val = $sayilar[$i];

$j = $i-1;

while($j>=0 && $sayilar[$j] < $val){

$sayilar[$j+1] = $sayilar[$j];

$j--;

}

$sayilar[$j+1] = $val;

}

return $sayilar;

}

echo "Original Array :\n";

echo implode(', ',$sayilar );

echo "\nSorted Array :\n";

print\_r(insertion\_Sort($sayilar));

$sure\_bitimi = microtime(true);

$sure = $sure\_bitimi - $sure\_baslangici;

echo "<br>Bekleme süresi: $sure saniye.\n";

//PHP kodlarına ayrılan belleğin miktarını bayt cinsinden döndürür.

echo 'Hafıza kullanımı: ',round(memory\_get\_peak\_usage()/1048576, 2), 'MB';

?>