**프로그래밍 실습 #4**

201501489 최영진

다음과 같은 정렬 알고리즘을 파이썬으로 구현하고, N 값의 변화에 따른 실행

시간을 측정한 후 실행 시간을 비교해 본다.

(1) 함수 naturalMergeSort(a[], n)를 사용하여 자연 합병 정렬을 구현하고, 합병

정렬과 성능을 비교 (연습문제 2.5 참조)

def mixed(arr, l, r, end):  
 tmp = 0  
 while arr[r] < arr[r-1]:  
 if arr[l]<=arr[r]:  
 l += 1  
 else:  
 tmp = r  
 while arr[l]>arr[tmp] and tmp <= end:  
 tmp += 1  
 arr[l], arr[tmp-1] = arr[tmp-1], arr[l]  
  
def mergeSort(arr, N):  
 i = 0  
 r = 0  
 while i < N-1:  
 left = i  
 while arr[i] <= arr[i+1]:  
 i += 1  
 r += 1  
 if (i < N-1):  
 i += 1  
 right = i  
 while arr[i]<=arr[i+1]:  
 i += 1  
 r += 1  
 mixed(arr, left, right, i)  
 i += 1  
 if r >= 2:  
 mergeSort(arr, N)  
  
def checkSort(a, n):  
 isSorted=True  
 for i in range(1, n):  
 if (a[i] > a[i+1]):  
 isSorted=False  
 if (not isSorted):  
 break  
 if isSorted:  
 print("정렬 완료")  
 else:  
 print("정렬 오류 발생")  
  
import random, time  
  
N = 1000  
a = []  
a.append(0)  
for i in range(N):  
 a.append(random.randint(1, N))  
a.sort()  
a.reverse()  
start\_time = time.time()  
mergeSort(a, N)  
end\_time = time.time() - start\_time  
print('자연 합병 정렬의 실행 시간 (N = %d) : %0.3f'%(N, end\_time))  
checkSort(a, N)

* 측정결과



* 비교

(2) 함수 tournamentSort(a[], n)를 사용하여 토너먼트 정렬을 구현하고, 히프 정

렬과 성능을 비교 (연습문제 2.8 참조)

def tournamentSort(a, n):  
 b = []  
 a\_height = int(math.log(n - 1, 2)) + 1  
 Max = int(math.pow(2, a\_height + 1))  
  
 for i in range(0, Max):  
 b.append(0)  
  
 b\_height = int(math.log(len(b) - 1, 2))  
 index = int(math.pow(2, b\_height))  
 for i in range(1, n + 1):  
 b[int(Max / 2) - 1 + i] = a[i]  
  
 while (b\_height) :  
 for k in range(index, index \* 2, 2):  
 if b[k] > b[k + 1]:  
 b[int(k / 2)] = b[k]  
 else:  
 b[int(k / 2)] = b[k + 1]  
 b\_height -= 1  
 index = int(index / 2)  
  
def checkSort(a, n):  
 isSorted = True  
 for i in range(1, n):  
 if (a[i] > a[i + 1]):  
 isSorted = False  
 if (not isSorted):  
 break  
 if isSorted:  
 print("정렬 완료")  
 else:  
 print("정렬 오류 발생")  
  
import time, random, math  
  
N = 2000000  
a = []  
a.append(0)  
for i in range(N):  
 a.append(random.randint(1, N))  
a.sort()  
a.reverse()  
start\_time = time.time()  
tournamentSort(a, len(a) - 1)  
end\_time = round((time.time() - start\_time), 3)  
print('토너먼트 정렬 실행 시간 (N = %d) : %0.3f' % (N, end\_time))

* 측정결과



* 비교