프로그래밍 실습 #3 보고서

Algorithm (월, 수)

채진석 교수님

201702797 채승민

모든 코드는 <https://github.com/carefreelife98/INU_Algorithm> 에도 존재합니다.

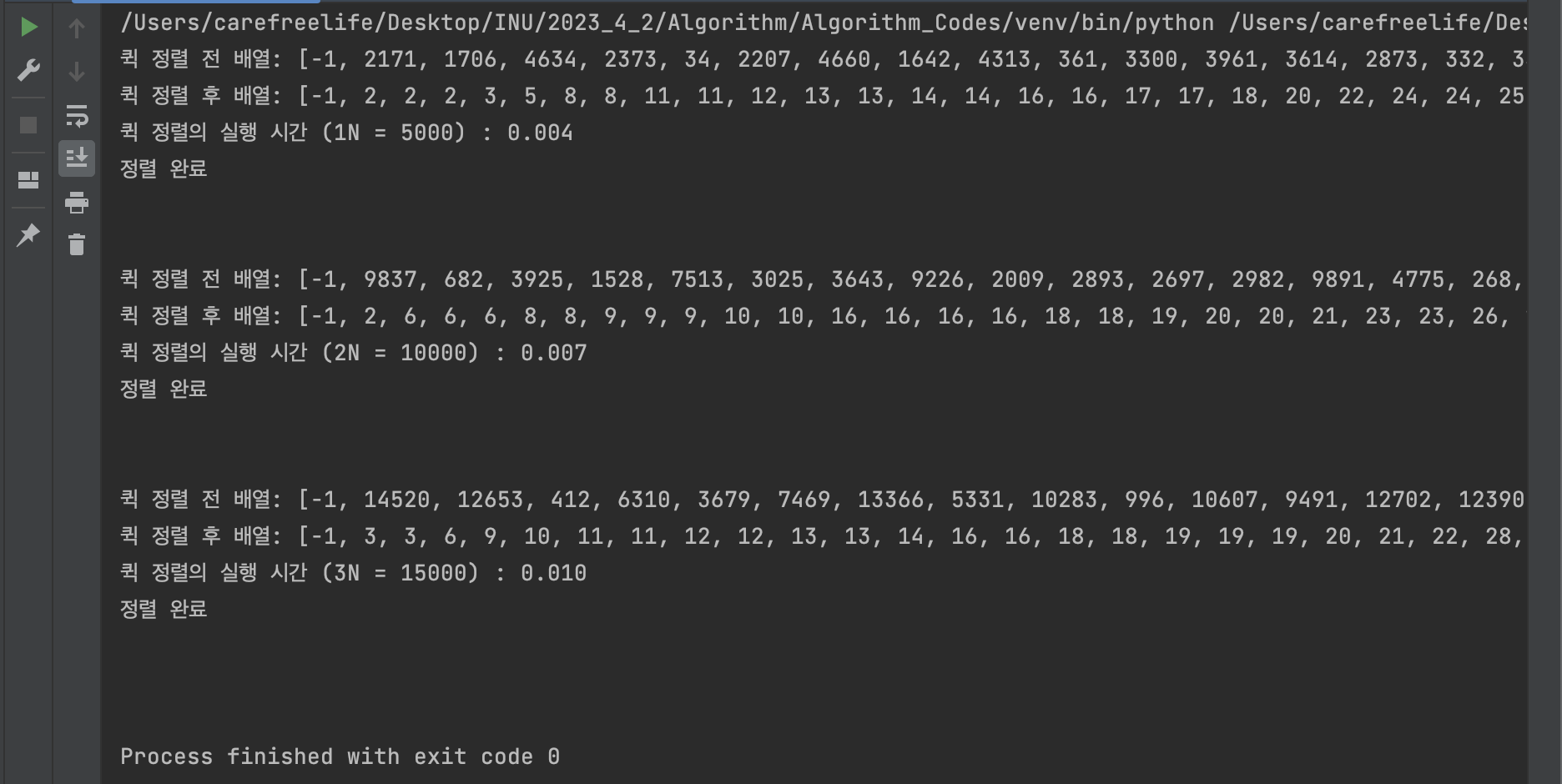
1. 퀵 정렬 알고리즘

ADL

|  |
| --- |
| quickSort(a, l, r) {  If (r > l) Then {  i <- Partition(a, l, r);  Call quickSort(a, l, i-1);  Call quickSort(a, i+1, r);  }  }  End quickSort  Partition(a, l, r) {  v <- a[r];  i <- l – 1;  j <- r;  While (1) {  i <- i + 1;  While (a[i] < v) {  i <- i + 1;  }  j <- j – 1;  While (a[j] > v) {  j <- j – 1;  }  If (i >= j) Then {  Break  }  a[i] 와 a[j] 교환;  }  a[i] 와 a[r] 교환;  Print(a);  Return i;  }  End Partition |

Test

* **N의 크기에 따른 실행시간의 변화 측정**

****

* **초기 데이터의 상태에 따른 실행 시간의 변화 측정**

**텍스트, 스크린샷, 컴퓨터, 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

1. 합병 정렬 알고리즘

ADL

|  |
| --- |
| merge(a, l, m, r)  i <- l;  j <- m + 1;  k <- l;  b <- CopyArray(a);    While (i <= m && j <= r) {  If (a[i] <= a[j]) Then {  b[k] <- a[i];  i <- i + 1;  Else {  b[k] <- a[j];  j <- j + 1;  }  End If  k <- k + 1;  }  End While    While (i <= m) {  b[k] <- a[i];  i <- i + 1;  k <- k + 1;  }  End While    While (j <= r) {  b[k] <- a[j];  j <- j + 1;  k <- k + 1;  }  End While    For (p <- l; p >= r; p++) {  a[p] := b[p]  }  End For  End merge  mergeSort(a, l, r) {  If (r > l) Then {  m <- (r + l) // 2;  Call mergeSort(a, l, m);  Call mergeSort(a, m + 1, r);  Call merge(a, l, m, r);  }  End If  }  End Algorithm |

Test

* **N의 크기에 따른 실행시간의 변화 측정**

**텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

* **초기 데이터의 상태에 따른 실행 시간의 변화 측정**

**텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명**

1. 히프 정렬 알고리즘

ADL

|  |
| --- |
| heapify(arr, n, i)  largest <- i;  left <- 2 \* i + 1;  right <-2 \* i + 2;  If left < n && arr[left] > arr[largest] Then {  largest <-left;  }  If right < n && arr[right] > arr[largest] Then {  largest <-right;  }  If largest ≠ i Then {  arr[i] 과 arr[largest] 교환;  Call heapify(arr, n, largest);  }  End If  End heapify  heapSort(arr)  n <-Length of arr;  For (i <- n // 2 – 1; i >= 0; i--){  Call heapify(arr, n, i);  }  For (i <- n – 1; i >= 1; i--){  arr[i] 와 arr[0] 교환;  Call heapify(arr, i, 0);  }  End heapSort |

Test

* **N의 크기에 따른 실행시간의 변화 측정**

**텍스트, 스크린샷, 블랙, 대수학이(가) 표시된 사진

자동 생성된 설명**

* **초기 데이터의 상태에 따른 실행 시간의 변화 측정**

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

1. Coding Test Practice #1

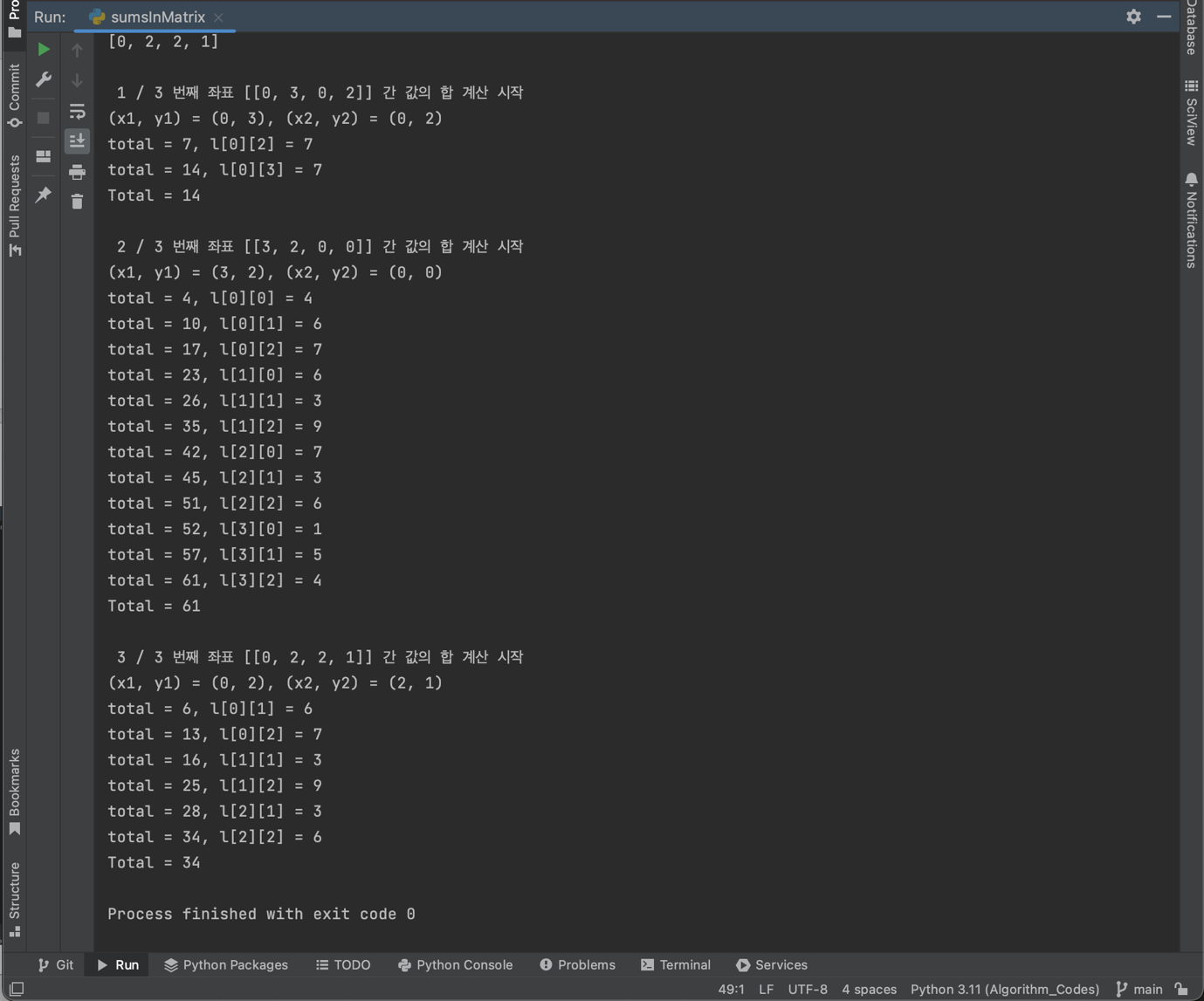
**Source Code**

import random  
def gen\_matrix(n, m):  
 try:  
 # (n,m) 범위 설정 확인  
 if 1 >= n >= 1024 or 1 >= m >= 100000:  
 return "1 <= n <= 1024, 1 <= m <= 100,000 범위가 잘못되었습니다."  
 print()  
 print(f"{n} X {n} 행렬 생성:")  
 a = []  
 for \_ in range(n):  
 b = []  
 for i in range(n):  
 num = random.randint(1, 9)  
 b.append(num)  
 a.append(b)  
 for i in a:  
 print(i)  
 return a  
 except Exception as e:  
 print(f"Error Occurred: [{e}]")  
  
def gen\_coordinate(a, n, m):  
 print(f"\n{n} X {n} 행렬에서 {m}개의 좌표 구간 합 결과를 반환:")  
  
 xy = []  
 for i in range(m):  
 temp = []  
 for j in range(4):  
 temp.append(random.randint(0, n - 1))  
 xy.append(temp)  
 for a in xy:  
 print(a)  
 return xy  
  
def sum\_mat\_bydiff(l, x1, y1, x2, y2):  
 total = 0  
 for i in range(x1, x2 + 1):  
 for j in range(y1, y2 + 1):  
 total += l[i][j]  
 print(f"total = {total}, l[i][j] = {l[i][j]}")  
 return total  
  
def sum\_result(matrix, coordinate: []):  
 try:  
 for idx, value in enumerate(coordinate):  
 print(f"\n {idx + 1} / {len(coordinate)} 번째 좌표 간 값의 합 계산 시작")  
 x1, y1, x2, y2 = 0, 0, 0, 0  
 for j in range(4):  
 match j:  
 case 0: x1 = value[j]  
 case 1: y1 = value[j]  
 case 2: x2 = value[j]  
 case 3: y2 = value[j]  
 print(f"(x1, y1) = ({x1}, {y1}), (x2, y2) = ({x2}, {y2})")  
 if x1 == x2 and y1 == y2:  
 print("두 좌표가 같습니다.")  
 total = matrix[x1][y1]  
 else:  
 min\_x, max\_x = min(x1, x2), max(x1, x2)  
 min\_y, max\_y = min(y1, y2), max(y1, y2)  
 total = sum\_mat\_bydiff(matrix, min\_x, min\_y, max\_x, max\_y)  
 print(f"Total = {total}")  
 except Exception as e:  
 print(f"Error Occurred: [{e}]")  
  
if \_\_name\_\_ == "\_\_main\_\_":  
 print("n, m 입력:")  
 n, m = map(int, input().split())  
  
 mat = gen\_matrix(n, m)  
 co\_ordinate = gen\_coordinate(mat, n, m)  
 sum\_result(mat, co\_ordinate)

**실행 모습**

텍스트, 스크린샷, 소프트웨어, 멀티미디어 소프트웨어이(가) 표시된 사진

자동 생성된 설명



**실행 방법**

1. 표의 크기 N과 합을 구해야 하는 횟수 M을 띄어쓰기로 구분하여 입력.
2. 이후 N x N 행렬과 두 좌표 (x1, y1) <-> (x2, y2) 및 그 횟수(M) 는 Random 으로 자동 생성됨.