**프로그래밍 실습 #2**

2023년 9월 2주차

201702797 채승민

1. 다음 점화식을 풀어라.
   * + - 1. , 이고 .

[풀이]

N >= 2

C\_2 = C\_1 +2

= 3

C\_3 = C\_2 +3

= C\_1 +2 +3

= 6

C\_4 = C\_3 +4

= C\_1 +2 +3 +4

= 1 +2 +3 +4

= 10

…

C\_N = C\_N-1 +N

= N(N+1)/2

= (N^2 +N)/2

= **O(N^2)**

* + - * 1. , 이고 .

[풀이]

N = 2인 경우: C\_2 = C\_(2/2) + 1 = C\_1 + 1 = 0 + 1 = 1

N = 4인 경우: C\_4 = C\_(4/2) + 1 = C\_2 + 1 = 1 + 1 = 2

N = 8인 경우: C\_8 = C\_(8/2) + 1 = C\_4 + 1 = 2 + 1 = 3

N = 16인 경우: C\_16 = C\_(16/2) + 1 = C\_8 + 1 = 3 + 1 = 4

N = 2^n 으로 가정 시,

C\_2^n = n + 1

따라서, **O(log N)**

(3) , 이고 .

텍스트, 친필, 문서, 폰트이(가) 표시된 사진

자동 생성된 설명

(4) , 이고 .

텍스트, 친필, 잉크, 공책이(가) 표시된 사진

자동 생성된 설명

(5) , 이고 .

텍스트, 친필, 폰트, 번호이(가) 표시된 사진

자동 생성된 설명

(6) , 이고 .

텍스트, 친필, 문서, 공책이(가) 표시된 사진

자동 생성된 설명

2. 다음 정렬 알고리즘을 ADL로 작성하고 파이썬으로 구현해 보라.

- 성능을 측정할 때 충분히 큰 N에 대해 N, 2N, 3N 등으로 입력 데이터의 개수를 증가시켜 가면서 실행 시간이 어떻게 변화하는지 측정함

- 또한, 랜덤 순서, 정렬된 순서, 역순으로 정렬된 순서로 데이터가 입력될 때의 실행 시간을 측정하여 입력 데이터의 순서에 얼마나 민감한지를 측정함

- 보고서를 만들 때 엑셀 등을 사용하여 실행 시간의 변화를 그래프로 나타냄

**(1) 선택 정렬**

--------------------------------------------------------------------

**ADL**

|  |
| --- |
| selection\_sort(a, n)  for (i <- 1; i <= n-1; i <- i+1) do {  min\_num <- i;  for(j <- i+1; j <= n; j <- j+1) do {  if (a[j] < a[min\_num]) then  min\_num <- j;  }  a[min\_num]과 a[i]를 교환;  }  end selection\_sort() |

--------------------------------------------------------------------

**N의 크기에 따른 실행시간의 변화 측정**

텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

자동 생성된 설명

**초기 데이터의 상태에 따른 실행 시간의 변화 측정**

텍스트, 스크린샷, 블랙이(가) 표시된 사진

자동 생성된 설명

**(2) 버블 정렬**

--------------------------------------------------------------------

**ADL**

|  |
| --- |
| bubble\_sort(a, n)  for(i <- n; i >= 1; i <- i-1) do  for(j <- 1; j <= i; j <- j+1) do  if(a[j] > a[j+1]) then  a[j]와 a[j+1]을 교환;  end bubble\_sort() |

--------------------------------------------------------------------

**N의 크기에 따른 실행시간의 변화 측정**

텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

자동 생성된 설명

**초기 데이터의 상태에 따른 실행 시간의 변화 측정**

텍스트, 스크린샷, 폰트, 대수학이(가) 표시된 사진

자동 생성된 설명

**(3) 삽입 정렬**

--------------------------------------------------------------------

**ADL**

|  |
| --- |
| insertion\_sort(l, n):  for(i <- 1; i <= n; i <- i+1) do {  key <- l[i];  j <- i-1;  while (j >= 0 && key < l[j]) do {  l[j + 1] <- l[j];  j <- j-1;  }  l[j + 1] <- key;  }  end insertion\_sort() |

--------------------------------------------------------------------

**N의 크기에 따른 실행시간의 변화 측정**

텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

자동 생성된 설명

**초기 데이터의 상태에 따른 실행 시간의 변화 측정**

텍스트, 스크린샷, 대수학이(가) 표시된 사진

자동 생성된 설명

**(4) 쉘 정렬**

--------------------------------------------------------------------

**ADL**

|  |
| --- |
| shell\_sort(arr, n)  gap <- n을 2로 나눈 몫;  while(gap > 0) do {  for(i <- gap; i <= n; i <- i+1) do {  temp <- arr[i];  j <- i;  while(j >= gap && arr[j - gap] > temp) do {  arr[j] <- arr[j - gap];  j <- j - gap;  }  arr[j] <- temp;  }  gap <- gap//2;  }  end shell\_sort() |

--------------------------------------------------------------------

**N의 크기에 따른 실행시간의 변화 측정**

**텍스트, 스크린샷, 폰트, 블랙이(가) 표시된 사진

자동 생성된 설명**

**초기 데이터의 상태에 따른 실행 시간의 변화 측정**

텍스트, 스크린샷이(가) 표시된 사진

자동 생성된 설명

**테스트 결과 보고서**

텍스트, 스크린샷, 번호, 그래프이(가) 표시된 사진

자동 생성된 설명