【 문제-1 】 (30점)

대표적인 작업 네트워크인 AOV(Activity on Vertex) 네트워크와 AOE (Activity on Edge) 네트워크에 관하여 다음의 물음에 답하시오.

(1) 다음과 같은 작업들로 구성된 프로젝트를 위한 AOV 네트워크와 AOE 네트워크를 각각 구성하시오. (10점)

정점 S: 프로젝트의 시작, 정점 E: 프로젝트의 종료

작업	시작조건	수행시간(주)	작업	시작조건	수행시간(주)
a1	프로젝트 시작	4	a6	a2 종료	6
a2	프로젝트 시작	3	a7	a2와 a3 종료	9
a3	프로젝트 시작	2	a8	a4와 a5 종료	2
a4	a1 종료	8	a9	a6 종료	5
a5	a2 종료	2	a7, a8, a9 모두 종료 시 프로젝트 종료		

모든 작업들은 위의 시작조건을 만족하면 병행하여 진행될 수 있다고 가정한다.

(2) 여러 개의 작업들로 구성된 프로젝트 스케줄을 관리하기 위한 자료구조로 AOE 네트워크가 AOV 네트워크보다 적합한 이유를 설명하시오. (5점)

(3) 문제 (1)의 결과로 구성된 AOE 네트워크에서 다음 각 물음에 답하시오. (10점)

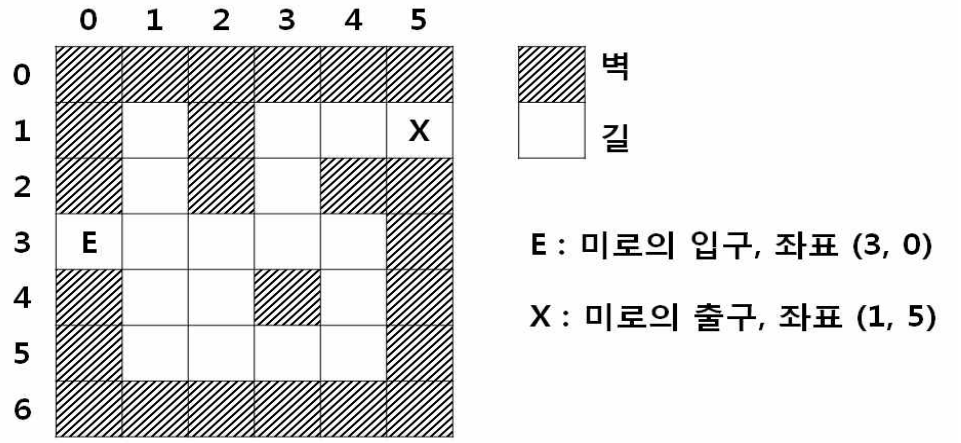
1) 각 작업에 대해 시작할 수 있는 가장 이른 시간(ET)과 프로젝트 수행을 지연하지 않으면서 가장 늦게 시작할 수 있는 시간(LT)을 구하시오.

2) 프로젝트를 완료하는데 필요한 최소 시간, 임계 경로, 임계 작업을 각각 구하시오.

(4) 문제 (1)의 결과로 구성된 AOE 네트워크에서 프로젝트 기간을 단축하기 위해 가속해야 하는 작업의 최소 개수 및 그에 해당하는 작업을 선택하여 제시하고, 최종적으로 단축한 프로젝트 최소 수행시간을 구하시오. (5점)

【 문제-2 】 (20점)

아래와 같이 미로가 주어졌을 때 다음의 물음에 답하시오.



(1) 스택의 특징을 설명하고, 미로 찾기에서 경로 정보를 저장하기 위해 스택을 활용하는 이유를 설명하시오. (5점)

(2) 미로의 입구부터 출구까지의 경로를 찾는 과정에서 스택을 이용한다. 다음 <조건>에 따라 출구를 찾을 때까지 아래 표 이후 부분을 완성하시오. (10점)

< 스택의 변화 과정 중 처음 8개 >

현재위치	(3,0)	(3,1)	(4,1)	(5,1)	(5,2)	(5,3)	(5,4)	(4,4)	...
이동방향	R	B	B	R	R	R	T	T	...
스택top	(3,0,T)	(3,1,R)	(4,1,R)	(5,1,T)	(5,2,T)	(5,3,T)	(5,4,^)	(4,4,^)	...
다음위치	(3,1)	(4,1)	(5,1)	(5,2)	(5,3)	(5,4)	(4,4)	(3,4)	...

< 조건 >

가) 미로의 입구는 E이며 출구는 X이다.

나) 각 위치에서 경로의 탐색은 왼쪽(L), 아래(B), 오른쪽(R), 위(T)의 순서로 이동하기로 한다. 단, 입구의 경우에는 R 방향부터 이동한다.

다) 현재위치가 (행,열)이고 이동방향이 D일 때 스택에 (행,열,ND)를 저장한다. ND는 나)에서 정의한 탐색순서에서 D 다음 순서를 의미한다. 예를 들어 (4,1)에서 B로 이동하는 경우, 스택에 B 다음 순서인 R을 사용하여 (4,1,R)을 저장하고, (3,3)에서 T로 이동하는 경우에는 T 다음 순서가 없으므로 ^기호를 사용하여 (3,3,^)를 저장한다.

(3) 주어진 미로에서 최적의 경로를 제시하고, 문제 (2)에서 찾은 경로가 최적의 경로에 비해 비효율적인 원인을 분석하고 이를 개선할 수 있는 방안을 제시하시오. (5점)

【 문제-3 】 (30점)

정렬 방식에 관하여 다음의 물음에 답하시오.

(1) 데이터 리스트의 키가 동일한 길이의 숫자나 문자열로 구성되어 있을 때, 키 값들 간의 비교를 하지 않고 큐(Queue) 형태의 추가적인 메모리를 사용하여 효과적으로 정렬할 수 있는 방식을 제시하고, 그 정렬 방식의 수행 시간을 시간복잡도(Time Complexity)로 설명하시오. (5점)

(2) 다음의 데이터 리스트에 대해 퀵(Quick) 정렬을 사용해서 오름차순으로 정렬하는 과정을 기술하시오. (단, 피벗(Pivot)은 리스트의 첫 번째 원소를 사용한다.) (5점)

[15, 27, 25, 5, 21, 10, 18, 23, 8]

(3) 셸(Shell) 정렬의 수행방법 및 장점을 삽입(Insertion) 정렬과 비교하여 설명하시오. (8점)

(4) 삽입 정렬과 합병(Merge) 정렬을 C 언어로 구현한 코드가 각각 다음과 같을 때, 두 정렬 방식의 시간복잡도를 최선의 경우(Best Case)와 최악의 경우(Worst Case)로 비교하여 설명하시오. (12점)

```
// 삽입 정렬
void insertion_sort(int list[], int n)
{  int i, j, key;
   for(i=1, i<n, i++) {
        key = list[i]; j = i;
        while((j>=0) && (list[j-1]>key)) {
           list[j] = list[j-1];
           j = j - 1;
        }
        list[j] = key;
   }
}
```

```
// 합병 정렬
typedef int element;
element sorted[MAX_SIZE};
void merge(element list[], int left, int middle, int right)
{  int i, j, k, t;
   i = left, j = middle + 1, k = left;
   while((i <= middle) && (j <= right)) {
      if (list[i] <= list[j]) {
         sorted[k] = list[i]; i++;
      } else {
         sorted[k] = list[j]; j++;
      }
      k++;
   }
   if (i > middle)
      for(t = j; t <= right; t++, k++)
         sorted[k] = list[t];
   else
      for(t = i; t <= middle; t++, k++)
         sorted[k] = list[t];
   for(t = left; t <= right; t++)
      list[t] = sorted[t];
}

void merge_sort(element list[], int left, int right)
{   int middle;
   if (left < right) {
      middle = (left + right) / 2;
      merge_sort(list, left, middle);
      merge_sort(list, middle+1, right);
      merge(list, left, middle, right);
   }
}
```

【 문제-4 】 (20점)

AVL 트리에 관하여 다음의 물음에 답하시오.

(1) 어떤 이진탐색 트리가 AVL 트리가 되기 위한 조건을 정의하시오. (단, 이진트리 T의 왼쪽 서브트리와 오른쪽 서브트리를 각각 T_L, T_R이라 하고, T_L과 T_R의 높이(Height)를 각각 h_L, h_R이라 한다.) (4점)

(2) AVL 트리에서 노드를 삽입할 때 트리의 불균형이 발생하면 균형 상태를 만들기 위해서 회전(Rotation) 작업을 수행해야 한다. 삽입시 불균형이 발생하는 4가지 경우를 열거하고, 각각의 경우에 대한 회전 방법을 설명하시오. (8점)

(3) 다음의 데이터를 순서대로 삽입하여 AVL 트리를 구성하시오. 또한 트리 구성 과정에서 불균형이 발생한 시점별로 적용된 회전 방법을 설명하시오. (8점)

14, 18, 23, 20, 27, 21, 5, 3