

## Lecture 11 : 큐(Queue)의 활용 - 대기(waiting) 문제

큐는 컴퓨터 운영체제에서 **process scheduling**에 가장 대표적으로 사용되는 자료구조이다. 현재 수행 중인 모든 **process**는 하나의 큐에 들어가 있고 CPU는 이 큐에서 프로세스를 꺼내서 일정 시간 동안 수행시킨 이후 다시 **queue**에 넣는다. 이렇게 **process**를 운영하는 방법을 **Time-Sharing System**이라고 부른다.

큐를 이용하여 어떤 작업을 처리할 때는 단순히 큐에 대상을 넣거나 빼는 작업 외 부가 정보를 관리하는 것이 매우 중요하다. 만일 시간적 순서(**temporal order**)가 아니라 특정 우선위에 따라서 가장 높은 우선순위의 대상을 먼저 처리할 수 있도록, 또는 이 작업이 효율적으로 가능하도록 구성된 또 다른 큐를 우선순위 큐(**priority queue**)라고 부른다. STL에서는 이 역시 제공하고 있으므로 실전을 통해서 그 사용법을 익혀야 한다.

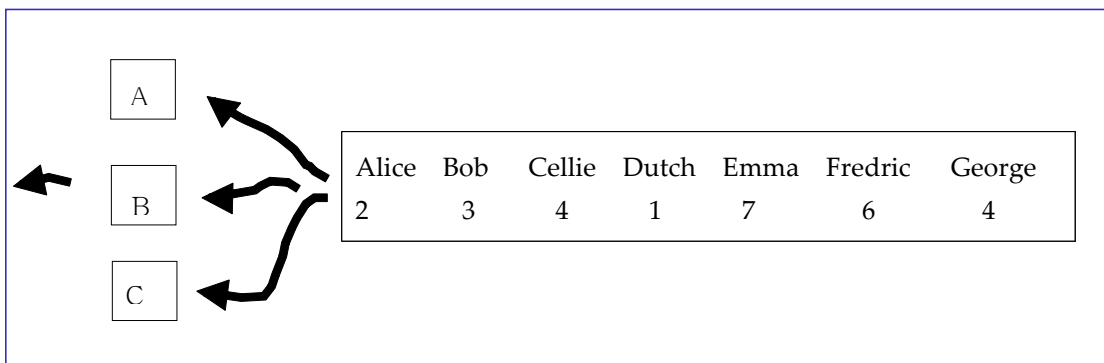
이와 같이 대기(**waiting**)이 존재하는 대부분 문제에서 **queue**가 사용된다. 이와 관련된 학문도 있는데 산업공학에서 다루는 **Queuing Theory**가 대표적인 분야이다.

11.1 STL adaptor인 큐(Queue)의 실제 내부 구조는 STL container인 **deque**(덱)로 구성되어 있다. **deque** = **double ended queue**. 이 디큐를 사용하기 위해서는 앞부분에 반드시 아래와 같이 선언해야 한다.

```
#include <deque>
```

11.2 **queue**안에 원소가 하나도 없을 경우에 **pop()**를 호출하면 어떤 오류가 발생되는지를 지금 사용하고 있는 컴파일러 기준으로 제시하시오.

11.3 다음 **Parallel Waiting** 큐 문제를 생각해보자. A, B, C는 미장원 의자 로 생각하자.



기차표(KTX)를 구입하려는 사람들이 줄을 서 있다. 그 순서는 입력파일 **ticket.inp**에 들어 있다. 파일의 각 줄에는 사람의 이름과 예상소요시간이 적혀있다. 그리고 역에는 3개의 판매 창구가 있다. 역에서는 이 3개의 창구에 사람들을 적절히 배분하고자 한다. 규칙은

다음과 같다.

- 모든 사람은 최초의 대기 큐에서 차례대로 기다린다.
- 현재 준비되어 있는 3개의 대기 큐 A, B, C 중에서 대기 시간이 짧은 쪽으로 들어간다고 가정하자.
- 만일 시간이 같다면 그 우선 순서는 A, B, C 로 들어간다.

11.4 여러분은 3개의 창구 Queue에 들어간 사람을 차례대로 출력하면 된다. 즉 ticket의 첫 줄에는 A, 둘째 줄에는 B, 마지막 줄에는 큐 C에 들어간 사람을 순서대로 출력한다.

| ticket.inp   | ticket.out                                   |
|--|--|
| 8<br>Alice 2<br>Bob 3<br>Cellie 4<br>Dutch 1<br>Emma 7<br>Fredreic 6<br>George 4 | Alice Dutch<br>Bob Emma<br>Cellie Frederic.. |

11.5 **〔도전문제〕** 실제 표를 사고 나오는 순서대로 사람을 계산하여 이것을 FinalQueue에 저장하는 작업을 해보시오. (사람들이 큐 사이를 오가는 시간을 무시할 정도로 작다고 생각한다. 즉 Alice가 창구-A로, Bob이 창구-B로, Cellie가 C 창구로 접근하는 시간은 무시할 정도로 작다고 가정한다.

11.6 **〔도전문제〕** 미용실 전용 의자가 3개뿐인 미용원이 있다. 많은 사람들이 인터넷으로 자신의 차례를 예약한다. 서비스받을 헤어 스타일에 따라 작업 시간이 정해진다. 단 이 작업은 적어도 하루 전에 모두 예약해야 한다. 여러분은 예약한 사람들에게 각각 도착해야 할 시간을 정해 알려주어야 한다. 단 모든 사람은 자신이 지정한 시간에서 2시간 이상 밀리면 예약 자체를 취소한다. 미용실의 입장에서 가장 많은 손님을 처리할 수 있는 프로그램을 Queue를 이용하여 작성해 보시오. 즉 하루는 12시간으로 잡고 가장 많은 손님을 “처리”하는 것이다. 단 서비스 비용은 동일하다고 가정한다.

| 의자<br>번호 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
|----------|---|---|---|---|---|---|---|---|---|----|----|----|
| 의자 1     |   |   |   |   |   |   |   |   |   |    |    |    |
| 의자 2     |   |   |   |   |   |   |   |   |   |    |    |    |
| 의자 3     |   |   |   |   |   |   |   |   |   |    |    |    |

### 11.7 문제 친구의 친구 (A Graph Problem)

모든 사람들에게는 친구가 있다. 그리고 그 친구들에게도 친구가 있다. 이 친구 관계를 이용해서 가장 사회성이 좋은 사람을 평가하고자 한다. 각 사람들의 친구관계는 다음과 같다. Friend[i][j]=1 이면 i가 j를 친구로 생각한다는 것이다. 즉  $i \rightarrow j$  이지만 그 역은 성립하지 않는다. 이들의 관계가 다음 행렬로 표시되어 있다.

$F(x, k)$ 는 특정인  $x$ 의  $k$ 친구 집합을 나타낸다.

$F(x, k) = F(F(x, k-1), 1)$  로 정의된다. 즉  $k$ 친구는  $k-1$  친구의 1차 친구.

$$F(1, 3) = \{ \quad \}$$

$$F(1, 4) = \{ \quad \}$$

$$F(7, 2) = \{ \quad \}$$

$$F(7, 3) = \{ \quad \}$$

$$F(12, 5) = \{ \quad \}$$

$$F(15, \infty) = \{ \quad \}$$

|    | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 | 13 | 14 | 15 |
|----|---|---|---|---|---|---|---|---|---|----|----|----|----|----|----|
| 1  | ○ |   |   |   |   |   | ○ |   | ○ |    |    |    |    |    |    |
| 2  |   | ○ |   | ○ |   |   |   |   |   |    |    |    | ○  |    |    |
| 3  |   |   | ○ |   |   |   | ○ |   |   | ○  |    | ○  |    |    |    |
| 4  |   |   |   | ○ |   |   |   |   |   |    |    |    | ○  |    |    |
| 5  | ○ |   |   | ○ | ○ |   | ○ |   |   |    |    | ○  |    |    | ○  |
| 6  |   |   |   | ○ |   | ○ |   |   |   |    |    | ○  |    |    |    |
| 7  |   | ○ |   |   |   |   | ○ |   |   |    |    |    |    |    |    |
| 8  |   |   |   |   |   |   |   | ○ |   |    |    |    |    | ○  |    |
| 9  |   |   |   | ○ |   |   |   | ○ | ○ |    |    | ○  |    |    |    |
| 10 |   |   |   |   |   |   |   |   |   | ○  |    |    |    |    |    |
| 11 |   | ○ |   | ○ |   |   | ○ |   |   |    | ○  |    |    |    |    |
| 12 |   |   |   | ○ |   |   |   |   | ○ |    |    | ○  |    |    |    |
| 13 |   |   |   |   | ○ |   |   |   |   |    |    |    | ○  |    |    |
| 14 |   |   | ○ |   |   |   | ○ |   |   |    |    |    | ○  | ○  |    |
| 15 | ○ |   |   |   |   |   |   | ○ |   |    |    |    |    |    | ○  |

어떤 특정인  $x$ 와 친구 관계로 연결된 모든 사람을 찾아내는 알고리즘은 다음과 같다. 시작 대상을  $s$ 라고 가정한다.

step 1) 초기화된 큐  $Q$ 를 준비한다.

step 2)  $Q$ 에 시작 노드  $s$ 를 추가한다. 다음부터는 Looping 구조이다.

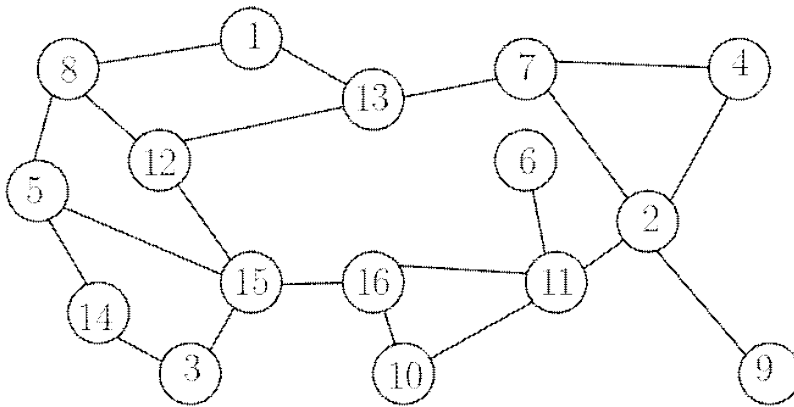
```

while( Q가 비어있지 않을 동안 ) {
    f ← Q.front( ) ;
    Q.pop( ) ;
    f와 연결된 사람 중에서, 이미 한번 이상 방문되지 못한 사람 h를
    순서대로 Q.insert( h ) 한다.
} // end of while

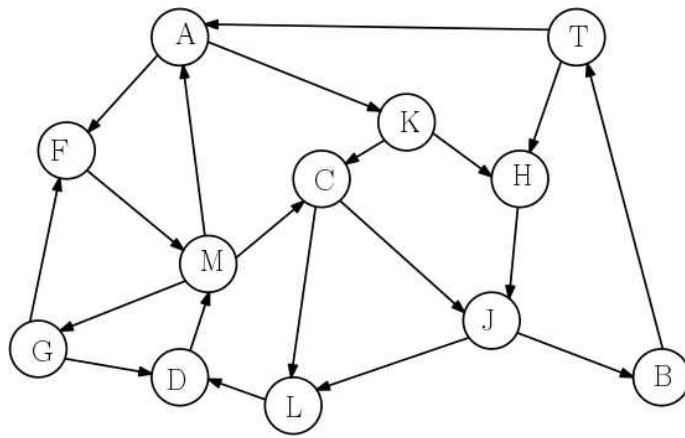
```

11.8 다음 친구 관계를 연결 행렬 (adjacency matrix) 로 나타내고  $F(x, k)$ 를 계산하시오.  
 다음 표에서 빈 칸을 채우시오.

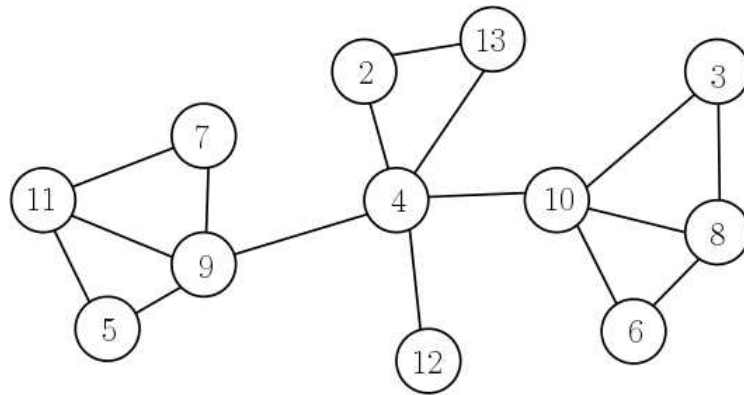
| k=0  | 1     | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 |
|------|-------|---|---|---|---|---|---|---|---|----|
| V_1  | 8, 13 |   |   |   |   |   |   |   |   |    |
| V_2  |       |   |   |   |   |   |   |   |   |    |
| V_5  |       |   |   |   |   |   |   |   |   |    |
| V_10 |       |   |   |   |   |   |   |   |   |    |
| V_4  |       |   |   |   |   |   |   |   |   |    |



11.9 생태계의 먹이사슬은 방향 그래프로 표현될 수 있다. 두 개체  $x, y$ 가 있을 때  $x$ 가  $y$ 의 먹이일 때 이것은 directed edge  $\overrightarrow{(x, y)}$ 로 나타낸다.  
 a) 이 방향 그래프에서 cycle이 있는지 확인하는 알고리즘을 제시하시오.



11.10 다음은 어떤 도시의 도로망이다. 여기에 우리는 하나의 소방서를 세우려고 한다. 가장 먼 것이 가장 짧은 “가운데”의 위치에 세워야 한다. 과연 이 가운데 위치인 **center**는 그래프의 어디에 있을까 ?



11.10 Queue 2개를 이용하여 입력되는 수열을 정렬하기 (가능 여부를 먼저 판단.)  
 두 개의 Queue에 서로 다른 숫자가 이미 저장되어 있다. 이 2군 데에 저장된 숫자를 queue의 pop( )만을 이용하여 새로운 queue에 오름차순으로 정렬(sorting)하는 과정을 제시하시오.

11.11 [중요] 2017, 2019 자료구조 중간고사 출제 문제

- 2개의 stack만을 이용해서 하나의 Queue를 simulation 하시오.
- 2개의 Queue만을 이용해서 하나의 Stack을 simulation 하시오.

약간의 보조 template 변수는 허용된다.

### 11.12 현미경 사진 분석 (바이러스 헤아리기)

- Connected component 찾아내기 (graph problem)
- 상하좌우, 4개의 이웃 셀은 중심 cell과 연결된 cell이다. - 4-connected model
- 어떤 경우 8-connected model이 사용되기도 한다.

|   |   |  |   |   |   |   |
|---|---|--|---|---|---|---|
|   |   |  |   |   |   | 4 |
|   |   |  | 1 |   | 4 | 4 |
|   |   |  | 1 | 1 |   |   |
|   |   |  | 1 | 1 |   |   |
|   | 3 |  |   |   |   |   |
| 3 | 3 |  |   | 2 | 2 |   |
|   | 3 |  |   |   | 2 | 2 |

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
|   |   |   |   |   | 1 |   |
|   | 1 | 1 | 1 | 1 |   |   |
|   |   | 1 |   | 1 |   |   |
|   |   | 1 | 1 | 1 |   | 2 |
|   |   | 1 |   |   |   | 2 |
|   |   |   |   |   | 2 | 2 |
| 3 |   |   | 2 | 2 |   |   |

- 11.13 코로나로 인하여 N명의 사람이 백신 접종을 위하여 줄을 서있다. 그런데 나이에 따라서 서로 다른 백신을 접종해야 한다. 20살 이하는 V1을 21살에서 60까지는 V2, 61 이상은 V3을 접종해야 한다. 전체 대기 Queue의 각 원소는 <이름,나이> pair 데이터가 주어져 있다. 즉 전체 사람은 3개의 독립된 Queue로 분산되어야 한다. 이렇게 접종을 할 때 각 큐에 들어간 사람을 순서대로 출력하는 프로그램을 작성하시오.

|    | 이름      | 나이 |
|----|---------|----|
| 1  | Brad    | 12 |
| 2  | Cellie  | 34 |
| 3  | Dempos  | 65 |
| 4  | Emile   | 43 |
| 5  | Fred    | 19 |
| 6  | Gomory  | 54 |
| 7  | Harris  | 63 |
| 8  | Idelia  | 9  |
| 9  | Jose    | 34 |
| 10 | Kosmid  | 27 |
| 11 | Lora    | 67 |
| 12 | Messi   | 11 |
| 13 | Nikkita | 15 |
| 14 | Operia  | 40 |
| 15 | Peter   | 73 |

| No. | V1<br>1-20 | V2<br>21-60 | V3<br>61- |
|-----|------------|-------------|-----------|
| 1   |            |             |           |
| 2   |            |             |           |
| 3   |            |             |           |
| 4   |            |             |           |
| 5   |            |             |           |
| 6   |            |             |           |
| 7   |            |             |           |
| 8   |            |             |           |