

# 추억 여행

최근 반복되는 일상에 지친 회사원 인하는 여행을 통해 기분 전환을 하려 한다. 평소 여행을 많이 다녔던 인하는 이번 여행은 새로운 방법으로 여행을 가고자 한다. 그 방법은 이전에 여행했던 서로 다른 지역들을 시간 순서대로 여행하는 방법이다. 인하는 이러한 방법으로 다니는 여행을 추억 여행이라 한다.

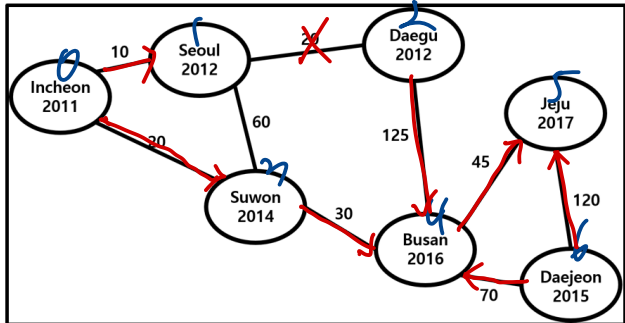
인하는 계획을 세우기 위해, 지도와 버스 시간표를 확인하며 필요한 정보를 노트에 작성하기로 하였다. 메모의 왼쪽에는 지역명과 여행했던 연도가 적혀있고 오른쪽에는 버스 노선 정보가 적혀있다. 버스 노선은 두 지역과, 그 지역 간의 이동 시간으로 주어지며, 두 지역 A, B 간에 버스 노선이 있으면, A에서 B로 또는 B에서 A로 이동이 가능하다.

계획을 세우는 인하는 추억 여행이라는 테마에 맞추어 다음 지역으로 이동할 때, 이전에 여행했던 연도를 기준으로 이동하기로 하였다. 즉, 2012년도에 여행한 지역 A와 2014년도에 여행한 지역 B가 있으면 A를 B보다 먼저 여행해야 한다. 그리고 인하는 지난 여행들에서 바빠서 보지 못했던 경치들을 즐길 수 있게, 이동 시간이 최대한 길도록 계획을 세울 것이다. 단, 선택된 모든 지역의 여행 연도는 달라야 한다. 인하를 도와 이동 시간이 최대한 길도록 여행을 계획하고, 그 때의 총 이동 시간을 알아보자.

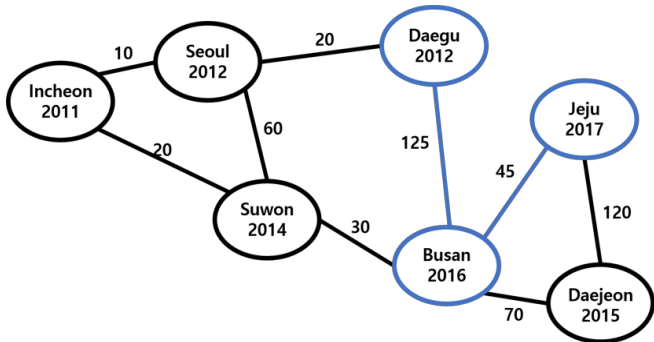
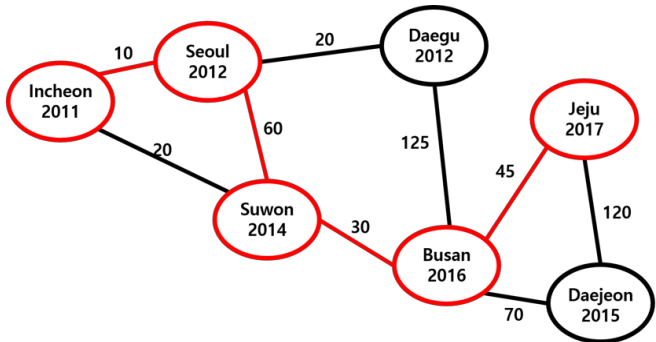
각 지역에는 충분히 많은 버스가 있어 바로 출발이 가능하고 버스를 타는 시간 외에 다른 시간은 고려하지 않으며, 모든 버스는 항상 일정한 속도로 이동한다고 가정한다.

예를 들어, 아래 그림은 인하가 작성한 메모와 이를 토대로 그린 그래프이다.

지역명	여행한 연도	지역 A	지역 B	이동 시간
Incheon	2011	Daegu	Seoul	20
Seoul	2012	Seoul	Suwon	60
Daegu	2012	Suwon	Incheon	20
Jeju	2017	Incheon	Seoul	10
Daejeon	2015	Daegu	Busan	125
Busan	2016	Suwon	Busan	30
Suwon	2014	Busan	Jeju	45
		Busan	Daejeon	70
		Daejeon	Jeju	120



만약, 아래 그림의 빨간색 경로(Incheon 출발, Jeju 도착)로 여행을 계획했을 경우, 인하의 총 이동 시간은 145이다. 하지만, 파란색 경로(Daegu 출발, Jeju 도착)로 여행을 계획했을 경우, 여행하는 지역의 수는 3개로 더 적지만 총 이동 시간은 170으로 빨간색 경로로 계획했을 때보다 더 많은 시간을 이동하게 된다.



인하가 작성한 메모가 주어졌을 때, 작성한 메모를 바탕으로 여행 가능한 최대 이동 시간을 출력한다.

※ 프로그램의 실행 시간은 1 초, 메모리 사용량은 512MB 를 초과할 수 없다.

사용할 수 있는 언어는 C, C++로 제한한다. C++의 경우 main 함수 내의 시작 지점에 다음 내용을 추가함으로써 cin, cout 의 입출력 속도를 개선할 수 있다.

```
ios_base::sync_with_stdio(false);
cin.tie(NULL);
cout.tie(NULL);
```

단, 위의 내용을 추가할 경우 cin, cout 만 사용해야 하며, scanf, printf 등 C 입출력을 혼용해서 사용하면 안된다. C++의 std::endl 의 경우 출력 속도가 느리므로, cout<<endl; 대신 cout<<"\n";을 사용하는 것을 권장한다.

## 입력

첫 번째 줄에는 테스트 케이스 수  $T$  ( $1 \leq T \leq 10$ )가 주어진다.

이후 각 테스트 케이스의 정보가 다음과 같이 주어진다.

- 인하가 여행했던 지역의 개수  $N$  ( $5 \leq N \leq 50,000$ )과 버스 노선의 개수  $M$  ( $5 \leq M \leq 100,000$ )이 주어진다.
- 이후  $N$ 개의 줄에 걸쳐, 지역명  $s$  ( $1 \leq |s| \leq 15$ )와 해당 지역을 여행한 연도  $y$  ( $1 \leq y \leq 1,000,000$ )가 공백으로 구분되어 주어진다.  $s$ 는 대소문자를 구분하는 영어 알파벳으로 구성된 문자열이며 중복된 값은 주어지지 않는다.
- 이후  $M$ 개의 줄에 걸쳐, 버스 노선들이 주어진다. 각 노선은 지역명  $s_1, s_2 (\in s)$ 와 이동 시간  $t$  ( $1 \leq t \leq 10,000$ )로 구성되어 있고,  $s_1, s_2, t$ 가 공백으로 구분되어 주어진다. 이는  $s_1$ 과  $s_2$  사이를  $t$ 시간에 이동하는 버스 노선이 존재한다는 것을 의미한다.

$n \times n \log n$  Same

## 출력

각 테스트 케이스마다 작성한 메모를 바탕으로 여행 가능한 최대 이동 시간을 한 줄에 출력한다.

### 예제 입출력

예제 입력	예제 출력
2	170
7 9	160
Incheon 2011	
Seoul 2012	
Daegu 2012	
Jeju 2017	

Daejeon 2015	0 2011, Incheon	0 [6, 1]
Busan 2016		
Suwon 2014	1 2012, Seoul	1 [6]
Daegu Seoul 20		
Seoul Suwon 60	2 2012, Daegu	2 [1, 5]
Suwon Incheon 20	3 2014, Suwon	3 [ ]
Incheon Seoul 10		
Daegu Busan 125	4 2015, Daejeon	4 [5, 3]
Suwon Busan 30		
Busan Jeju 45	5 2016, Busan	5 [3]
Busan Daejeon 70		
Daejeon Jeju 120	6 2017, Jeju	6 [5]
10 15		
Ava 2001		0
Akyab 2002		
Bagan 2002		
Bago 2003		
Mandalay 2004		
Mrawk 2005		
Naypyidaw 2005		
Prome 2006		
Toungoo 2007		
Yangon 2008		
Ava Akyab 15		
Ava Yangon 80		
Akyab Bagan 10		
Akyab Bago 20		
Bagan Mandalay 20		
Bagan Mrawk 60		
Bago Mrawk 20		
Bago Yangon 50		
Mandalay Prome 65		
Naypyidaw Prome 70		
Naypyidaw Toungoo 120		
Prome Toungoo 45		
Akyab Mrawk 20		
Bago Prome 80		
Mrawk Prome 30		

Daejeon 2015 Busan 2016 Suwon 2014 Daegu Seoul 20 Seoul Suwon 60 Suwon Incheon 20 Incheon Seoul 10 Daegu Busan 125 Suwon Busan 30 Busan Jeju 45 Busan Daejeon 70 Daejeon Jeju 120 10 15 Ava 2001 Akyab 2002 Bagan 2002 Bago 2003 Mandalay 2004 Mrawk 2005 Naypyidaw 2005 Prome 2006 Toungoo 2007 Yangon 2008 Ava Akyab 15 Ava Yangon 80 Akyab Bagan 10 Akyab Bago 20 Bagan Mandalay 20 Bagan Mrawk 60 Bago Mrawk 20 Bago Yangon 50 Mandalay Prome 65 Naypyidaw Prome 70 Naypyidaw Toungoo 120 Prome Toungoo 45 Akyab Mrawk 20 Bago Prome 80 Mrawk Prome 30	
---	--