CSE2011 Problem Solving, 2016 Spring

2013312343 이상헌

**Homework 5-2**

**1. 문제 이해**

(1) 문제

**Problem 5-2: Wire 2 (40 points)**

- There is a 100-story building. In this building, there are N electric wires in the wall. We can see each wire in either the first or the 100th story. Since the shape and color of every wire is the same, we cannot match the connectivity of wires in the first story and those in the 100th story.

- In detail, we mark numbers from a1 to aN to each wire in the first story, and also mark numbers from b1 to bN to each wire in the 100th story. We want to find each wire’s connectivity between aX and bY. There are three operations we can do.

O1. We can bind/untie two wires. (This can only be done in either the first or the 100th story.)

O2. We can test whether two wires are connected or not. (This can only be done in either the first or the 100th story.)

O3. We can move from the first story to the 100th story, or from the 100th story to the first story.

- Now, an elevator is installed in the building. However, you missed a tester for O2; you need to borrow a tester. You will pay 1 million won for each operation for O2, so you should minimize the number of operations of O2. Describe your algorithm to find connectivity of every wire while minimizing the number of operations of O2.

(2) 문제 정의

- 100층의 건물 꼭대기와 바닥에 N개의 도선이 노출되어 있다.

- 꼭대기와 바닥에 노출된 각각의 N개의 도선의 연결 상태는 미지이다. 바닥의 노출된 각 도선을 a1 - aN이라 하고, 꼭대기에 노출된 각 도선을 b1 - bN이라 하자.

- 두 도선을 연결 혹은 연결을 끊거나(O1), 두 도선의 연결상태를 확인하거나(O2), 건물의 꼭대기와 바닥을 오르내릴 수 있다.(O3)

- 두 도선의 연결 상태를 확인하는 횟수(O2 수)를 최소화하여 모든 도선의 연결 상태를 확인하는 알고리즘을 구현하라.

**2. 문제 해결**

*한 개를 제외한 나머지 도선 연결.*

- 도선의 연결 상태는 최상의 경우 혹은 최악의 경우가 될 수 있다. 모든 경우에 대해서 확실하게 도선의 연결 상태를 알 수 있는 경우를 우선으로 생각해보았다. 본 아이디어는 도선의 연결 상태를 하나씩 확인하며, 이미 연결 상태를 알고 있는 도선을 제외하는 방법이다.

- 아이디어

바닥에서 하나의 도선을 제외한 나머지 도선을 서로 이어지도록 연결한다. 이후 꼭대기로 이동하여 기준 도선을 하나 잡고, 기준 도선과 나머지 도선들 과의 연결 상태를 확인한다. 이 과정에서 연결되지 않았다고 나타나는 도선 쌍이 반드시 하나 존재할 것이다. 이것은 기준 도선 혹은 다른 도선이 바닥에서 연결에 제외시킨 하나의 도선과 연결되었다는 것을 알 수 있다.

이후 바닥으로 이동하여 연결 상태를 모두 끊고, 이미 연결 상태를 알고 있는 한 개의 도선을 제외한 나머지 도선들에 대해 같은 방법을 반복한다. 즉, 나머지 N-1개의 도선들 중에 하나의 도선을 제외한 나머지 도선을 서로 이어지도록 연결하고, 꼭대기로 이동하여 연결 상태를 확인하는 방법을 반복한다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | b10 |
|  |  |  |  |  |  |  |  |  |  |
| a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | a10 |

Table 1. N = 10인 경우에 대한 도선의 연결 상태.

예를 들어 N = 10인 경우를 생각해보자. 꼭대기와 바닥에 각각 10 개의 도선이 노출되어 있을 때, 각 도선의 연결 상태가 Table 1과 같다고 생각해보자.

첫 번째로, 바닥의 가장 끝에 있는 a10을 제외한 나머지 a1 – a9의 도선들을 서로 이어지도록 연결한다. 이는 a1과 a2, a2와 a3, …, a8과 a9를 연결한다. 이후 꼭대기로 올라가 기준 도선을 b1으로 잡고, 나머지 b2 – b10의 도선과의 연결 상태를 확인한다. 즉, (b1, b2), (b1, b3), …, (b1, b9)의 연결 상태를 확인한다.

연결 상태를 확인하는 중에는 크게 두 가지의 경우가 나타난다. 첫 번째 경우로 (b1, b2)가 연결되지 않은 경우이다. 이 경우에는 (b1, b3)의 연결 상태를 확인한다. (b1, b3)이 연결되어 있는 경우, b2가 a10과 연결되어 있다는 것을 의미하고, (b1, b3)이 연결되지 않은 경우, b1이 a10과 연결되어 있다는 것을 의미한다. 이는 Table 2와 같이 경우가 나뉜다.

|  |  |  |
| --- | --- | --- |
| (b1, b2)가 연결되지 않음. |  |  |
|  | (b1, b3)가 연결되어 있음. | (b1, b3)가 연결되지 않음. |
|  | a10과 b2가 연결. | a10과 b1이 연결. |

Table 2. (b1, b2)가 연결되지 않은 경우.

두 번째 경우로 (b1, b2)가 연결되어 있는 경우이다. 이 경우에는 (b1, b3), (b1, b4), …, (b1, b9) 순서로, 연결되지 않은 단 하나의 도선 쌍을 찾을 때까지 연결 상태를 확인한다. 예시에서는 (b1, b9)이 연결되지 않았다는 결과를 나타내고, 이는 b9이 a10과 연결되어 있다는 것을 의미한다. 또한 (b1, b9)까지 연결 상태가 모두 연결되었다는 결과가 나타나면, b10이 a10과 연결되어 있다는 것을 의미한다.

이후 바닥으로 이동하고, a10을 제외한 나머지 a1 – a9를 대상으로 위와 같은 방법을 반복한다. 즉, a9를 제외한 a1 – a8을 서로 이어지도록 연결하고, 꼭대기로 올라가 검사를 시행한다. 꼭대기에서의 검사 또한 b1 – b10들 중에 a10과 연결 되어있는 도선을 제외한 나머지 도선들 중에서 같은 방법으로 기준 도선을 잡고, 검사를 시행한다. 이와 같은 방법으로 모든 도선에 대해 하나씩 연결 상태를 확정 짓고, 제외하면서, 전체적인 연결 상태를 확인한다.

마지막에 a1과 a2가 남은 상황에 대해서 연결 상태를 확인하는 방법은 기존과 약간 다르다. 만일 a1과 a2, b1과 b2가 남았다고 가정해보자. 바닥에서 a1을 a10과 연결하고, 꼭대기로 올라가 b1과 b10의 연결 상태를 검사한다. 만일 연결이 되어있다면 a1과 b1이 연결되어 있다는 것을 의미하고, 연결되어 있지 않다면, a1과 b2가 연결되어 있다는 것을 의미한다.

- O2 수

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| b1 | b2 | b3 | b4 | b5 | b6 | b7 | b8 | b9 | b10 |
|  |  |  |  |  |  |  |  |  |  |
| a1 | a2 | a3 | a4 | a5 | a6 | a7 | a8 | a9 | a10 |

도선의 개수가 N개인 건물의 모든 도선의 연결 상태를 본 알고리즘을 기반으로 확인하고 싶은 경우에서 필요한 O2의 수를 구해보자. 극단적인 경우에는 도선이 Table 3과 같은 연결 상태를 나타낸 경우이다.

Table 3. N = 10인 경우, 극단적인 도선의 연결 상태.

Table 3과 같이 도선이 연결되어 있는 경우에 대해 살펴보자. 첫 번째로 a10을 제외시키고 a1 – a9를 연결한 후, 꼭대기로 이동하여 검사할 때에는 (b1, b2), (b1, b3), …, (b1, b9) 까지 모든 검사를 해야하므로 N-2번의 O2 가 필요하다. 이후 바닥에서 a10을 제외하고, 나머지 a1 – a9들 중 a9를 뺀 a1 – a8을 연결한 후에, 꼭대기로 이동하여 검사를 시행한다. 이 때 필요한 검사 수는 N-3번이다. 즉, O2 수는 1씩 줄어들면서, 1번까지 수행된다. 마지막에 a1과 a2가 남은 경우에 대해서도 추가적으로 1번의 O2 수행이 필요하다.

이러한 극단적인 경우에 대해서, 도선 간의 연결 상태를 확인하기 위해 필요한 O2 수는 다음 수식과 같이 계산된다. . 따라서 도선의 개수가 N개인 건물에서 각 도선의 연결 상태를 확인하는 데에 필요한 O2의 수는 N(N-3)/2 번이다. 즉, N=10인 경우에는 35번의 O2 수행이 필요하다. 만일 도선의 연결 상태가 극단적인 경우가 아니라면, 필요한 O2 수는 이보다 적다.

- 이점

본 알고리즘의 큰 장점은 정확성이다. 도선의 연결 상태에 상관 없이 정확한 결과를 도출할 수 있다. 또한 N개의 각 도선에 대해, 1번씩 만의 검사를 통해 한 도선의 연결되어 있는 반대편의 다른 도선을 알 수 있기 때문에 O(N)의 time complexity를 나타내는데, 이는 가장 효율적인 알고리즘이라고 생각한다. 전체적으로 N개의 도선의 모든 연결 상태를 확인하기 위해서는 O(N^2)의 time complexity를 나타낸다. 또한 극단적인 경우가 아닌 모든 다른 경우에 대해서 필요한 O2 수는 N(N-3)/2번 보다 항상 적게 나타난다.

- 한계

본 알고리즘의 한계로는 결국 효율성이다. 만일, 더 적은 O2 수로 정확한 결과를 도출할 수 있는 알고리즘이 존재한다면, 본 알고리즘의 한계가 발생하는 것이다. 하지만 time complexity 측면에서 분석한 결과는 본 알고리즘보다 더욱 효율적인 알고리즘은 존재하지 않을 것으로 예상한다.

**3. 고찰**

문제는 O2 수를 기준으로, 가장 효율적으로 건물 내의 도선의 연결 상태를 확인하는 알고리즘을 구축하는 것이다. 본 알고리즘은 바닥의 한 도선을 제외한 나머지 도선을 서로 연결하고, 꼭대기에서 하나씩 검사하여 제외한 도선과 대응하는(연결 되어있는) 도선을 찾고, 제외하여 같은 방법을 반복하는 기법이다. N개의 도선이 있는 건물에서 모든 도선의 연결 상태를 확인하는 데에 필요한 O2 수는, 가장 극단적인 경우에 N(N-3)/2 번이다. 일반적인 경우에는 이보다 적은 O2 수만으로도 정확한 결과를 도출할 수 있기 때문에 충분히 효율적인 알고리즘이라고 판단되었다.

**4. 참고문헌 및 사용**

- 이진규 교수님, [문제해결기법] CSE2011\_2016spring\_Lecture\_Note07, 2016