CSE2011 Problem Solving, 2016 Spring

2013312343 이상헌

**Homework 5-1**

**1. 문제 이해**

(1) 문제

**Problem 5-1: Wire 1 (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.

- There is no elevator in the building, so you should minimize the number of operations of O3. Describe your algorithm to find connectivity of every wire while minimizing the number of operations of O3.

(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)

- 건물의 꼭대기와 바닥을 오르내리는 횟수(O3 수)를 최소로 하여 모든 도선의 연결 상태를 확인하는 알고리즘을 구현하라.

**2. 문제 해결**

*이웃한 두 도선 연결 및 확인.*

- 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 정렬 알고리즘 중 Bubble Sort 알고리즘에서 착수하였다.

- 아이디어

바닥의 이웃한 두 도선을 연결한다. 이후 꼭대기로 이동하여 모든 도선들 중 임의의 두 도선의 연결 상태를 모두 확인한다. 이미 바닥의 두 도선이 연결되어 있기 때문에, 꼭대기의 임의의 두 도선 쌍들 중에 오직 하나만이 연결되었음을 알 수 있다. 이를 기록한다.

이후 바닥으로 이동하여 연결 되어있던 두 도선의 연결을 해제하고, 그 다음의 이웃한 두 도선을 연결하여 위와 같은 방법을 반복한다. 이후 기록한 모든 도선 쌍들의 연결 상태를 비교하여, 모든 도선의 각 연결 상태를 분석할 수 있다.

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

Table . 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 이웃한 두 도선인 a1과 a2를 연결한다. 이후 꼭대기로 이동하여 모든 b1 – b10의 도선들 중 임의의 두 도선의 연결 상태를 체크하여, 대응되는 두 도선 쌍을 찾는다. 즉, (b1, b2), (b1, b3), (b1, b4), …, (b1, b10), (b2, b3), (b2, b4), ..., (b2, b10), …, (b9, b10)의 도선 쌍의 연결 상태를 모두 확인하면, 이들 중 오직 하나의 도선 쌍만이 연결되어 있을 것이다. 예시의 경우에는 (b1, b10)이 연결되어 있고, 나머지는 연결되어있지 않을 것이다. 즉, (a1, a2)와 대응되는 도선은 (b1, b10)이다.

두 번째로, 바닥으로 내려가 a1과 a2의 연결을 끊고, a2와 a3를 연결한다. 이후 위와 같은 방법으로, 꼭대기의 연결된 두 도선 쌍을 찾는다. (a2, a3)와 대응되는 도선은 (b1, b2)이다. 위와 같은 두 단계를 반복하여, 바닥의 모든 이웃한 도선과 대응되는 꼭대기의 도선 쌍을 찾는다. 반복한 결과는 Table 2와 같이 기록된다.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| 바닥 쌍 | (a1,a2) | (a2, a3) | (a3, a4) | (a4, a5) | (a5, a6) | (a6, a7) | (a7, a8) | (a8, a9) | (a9, a10) |
| 대응 쌍 | (b1, b10) | (b1, b2) | (b2, b3) | (b3, b4) | (b4, b5) | (b5, b6) | (b6, b7) | (b7, b8) | (b8, b9) |

Table . Table 1과 같이 연결된 도선들의 연결 쌍

이제 Table 2와 같이 대응되는 도선 쌍을 토대로 모든 도선의 연결 상태를 알 수 있다. 이는 이웃한 두 개의 도선 쌍들을 분석하면 가능하다. (a1, a2)와 (a2, a3)에 대응되는 도선 쌍은 (b1, b10)과 (b1, b2)이다. 두 도선 쌍들 중에 겹치는 도선을 생각하면, a2와 b1이다. 즉, a2는 b1과 연결되어 있다는 것을 알 수 있다. 이와 같이 이웃한 두 도선 쌍들에 겹치는 도선을 확인하면서, 모든 도선의 연결 상태를 확인할 수 있다.

- O3 수

도선의 개수가 N개인 건물의 모든 도선의 연결 상태를 본 알고리즘을 기반으로 확인하고 싶은 경우에서 필요한 O3의 수를 구해보자. 일반적인 경우는 꼭대기에서 바닥으로 이동하여 바닥의 이웃한 두 도선을 연결하고, 다시 꼭대기로 이동하여 임의의 두 도선의 연결 상태를 검사한다. 즉, 대응되는 두 도선 쌍들을 찾기 위해서는 2번의 이동이 필요하다. 단, 처음 건물의 바닥에서 시작한다고 가정하면, 바닥의 도선들 중 이웃한 두 도선을 연결한 후에 꼭대기로 이동하여 연결 상태를 검사하기 때문에, 처음은 1번의 이동만이 필요하다.

따라서 도선의 개수가 N개인 건물의 모든 도선의 연결 상태를 알기 위해 필요한 O3 수는 다음 수식과 같다. . 즉, 도선의 개수가 N개인 경우, 필요한 O3 수는 2N -3 번이다. 예를 들어 N = 10인 경우에는 17번의 이동이 필요하다.

- 이점

본 알고리즘의 큰 장점은 정확성이다. 본 알고리즘을 이용한다면, 도선의 연결 상태에 상관 없이 모든 도선의 연결 상태를 정확히 알 수 있다. 즉, 아무리 극단적인 경우에 대해서도 정확한 결과를 도출할 수 있다. 또한, 본 알고리즘에서 필요한 O3의 수는 2N -3 번으로, O(N)의 time complexity를 나타낸다. 이는 다른 알고리즘들과 비교하였을 때, 충분히 효율적으로 결과를 도출할 수 있다는 것을 의미한다.

- 한계

본 알고리즘의 가장 큰 한계는 도선의 연결 상태에 따라 O3 수를 줄일 수 있는 알고리즘이 존재하는 경우 나타난다. 본 알고리즘의 O3 수는 전적으로 도선의 수에 따라 좌우된다. 만일 도선의 연결 상태를 모두 확인하는 데에 필요한 O3 수가 도선의 연결 상태에 따라 좌우되는 알고리즘이 존재한다면, 최적화된 연결 상태인 경우에 대해 본 알고리즘보다 더욱 효율적으로 결과를 도출할 수 있을 것이다.

**3. 고찰**

문제는 O3 수를 기준으로, 가장 효율적으로 건물 내의 도선의 연결 상태를 확인하는 알고리즘을 구축하는 것이다. 본 알고리즘은 이웃한 두 도선을 연결하여 대응하는 반대편의 도선 쌍을 기록하고, 이웃하다는 점을 기반으로 대응하는 두 도선 쌍들 중 겹치는 도선을 통해 연결 상태를 확인하는 방법이다. N개의 도선이 건물 내에 존재하는 경우, 본 알고리즘을 통해 결과를 도출할 때 필요한 O3 수는 2N -3 번이다. 본 알고리즘은 어느 경우에 대해서도 정확한 결과를 도출할 수 있지만, 특정 경우에 대해서 O3 수를 줄일 수 있는 방안에 대해서는 포함되지 않았다는 한계점을 가지고 있다. 만일 최적의 도선 연결 상태를 가지고 있을 때, 즉 검사 중에 결과에 대한 윤곽이 잡힐 수 있다면, 총 필요한 O3의 수를 줄일 수 있을 것이라고 생각한다.

**4. 참고문헌 및 사용**

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