JavaScript 백트래킹 알고리즘

백트래킹 문제 풀이

백트래킹 문제 풀이 | 코딩 테스트에서 자주 등장하는 백트래킹 알고리즘 이해하기

강사 나동빈

JavaScript
백트래킹 알고리즘

백트래킹 문제 풀이

외판원 순회 2: https://www.acmicpc.net/problem/10971

JavaScript
백트래킹
백트래킹
문제 풀이

혼자 힘으로 풀어보기

문제 제목: 외판원 순회 2

문제 난이도: ★★★☆☆

문제 유형: 백트래킹, 경우의 수, 완전 탐색

추천 풀이 시간: 50분

JavaScript
백트래킹
백트래킹
문제 풀이

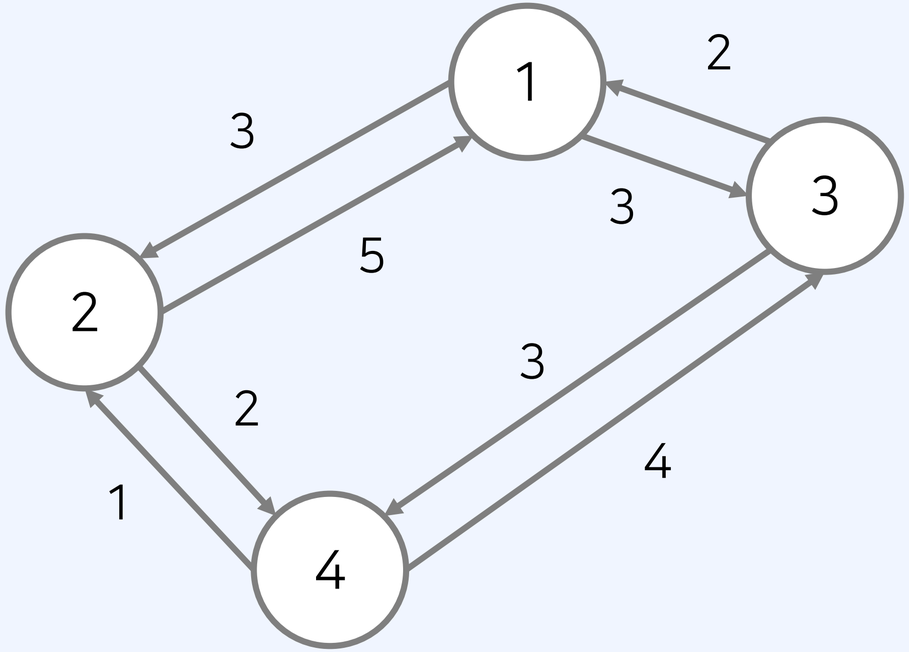
문제 해결 아이디어

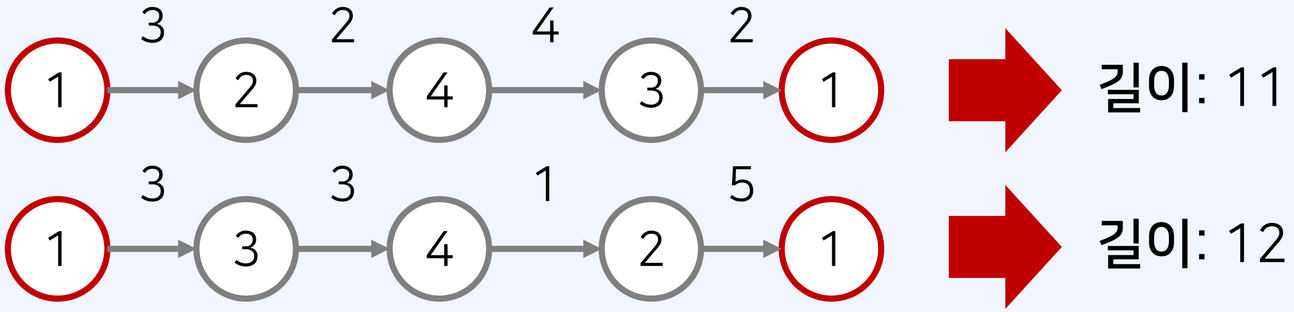
- 본 문제는 **외판원 순회**(traveling salesman problem, TSP) **문제**다.
- 어느 한 도시에서 출발해 N개의 도시를 모두 거쳐 다시 원래의 도시로 돌아와야 한다.

외판원 순회 2: https://www.acmicpc.net/problem/10971

문제 해결 아이디어

- 예시를 보면 다음과 같다.
- **1번 노드**에서 출발한다면?





외판원 순회 2: https://www.acmicpc.net/problem/10971

JavaScript 백트래킹
백트래킹 문제 풀이

정답 코드 예시

```
// 2부터 N까지의 수를 하나씩 골라 나열하는 모든 순열을 계산
function dfs(depth) {
  if (depth == n - 1) { // 현재 순열에 대한 처리
    let totalCost = 0; // 1번 노드에서 출발
    let cur = 1; // 1번 노드에서 출발
    for (let i = 0; i < n - 1; i++) { // 현재 순열에 따라서 노드 이동
      let nextNode = result[i]; // 다음 이동 노드까지의 비용 계산
      let cost = graph[cur][nextNode];
      if (cost == 0) return; // 이동 불가능하면 무시
      totalCost += cost; // 이동 가능하면, 비용을 더하고 노드 이동
      cur = nextNode;
    }
    // 마지막 노드에서 1로 돌아오는 것이 필수
    let cost = graph[cur][1];
    if (cost == 0) return; // 이동 불가능하면 무시
    totalCost += cost; // 이동 가능하면, 비용을 더하고 노드 이동
    minValue = Math.min(minValue, totalCost); // 경로의 최소 비용 저장
  }
  for (let i = 2; i <= n; i++) {
    if (visited[i]) continue;
    result.push(i); // 방문 처리
    visited[i] = true;
    dfs(depth + 1); // 재귀 함수 호출
    result.pop(); // 방문 처리 해제
    visited[i] = false;
  }
}
```

정답 코드 예시

```
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let n = Number(input[0]);
let graph = []; // 전체 그래프(graph) 정보 입력
for (let i = 0; i <= n; i++) graph.push([0]);
for (let i = 1; i <= n; i++) {
  line = input[i].split(' ').map(Number);
  for (let j = 0; j < n; j++) graph[i].push(line[j]);
}
let visited = new Array(11).fill(false); // 방문 처리 배열
let result = []; // 순열 정보 배열
let minValue = 1e9;

dfs(0);
console.log(minValue);
```

도영이가 만든 맛있는 음식: https://www.acmicpc.net/problem/2961

혼자 힘으로 풀어보기

JavaScript
백트래킹
백트래킹
문제 풀이

문제 제목: 도영이가 만든 맛있는 음식

문제 난이도: ★★★☆☆

문제 유형: 백트래킹, 경우의 수, 완전 탐색

추천 풀이 시간: 50분

문제 해결 아이디어

JavaScript
백트래킹
백트래킹
문제 풀이

- 재료로 A, B, C, D 네 가지가 있다고 가정하자.
- 문제의 요구사항은 **모든 길이**에 대한 가능한 **모든 조합**을 구하는 것이다.
- 그 이유는 재료 (A, B, C)를 사용한 것과 (A, C, B)를 사용한 것이 같은 결과를 가지기 때문이다.

1. **길이 #1**: (A), (B), (C), (D)
2. **길이 #2**: (A, B), (A, C), (A, D), (B, C), (B, D), (C, D)
3. **길이 #3**: (A, B, C), (A, B, D), (A, C, D), (B, C, D)
4. **길이 #4**: (A, B, C, D)

도영이가 만든 맛있는 음식: https://www.acmicpc.net/problem/2961

정답 코드 예시

JavaScript
백트래킹
백트래킹
문제 풀이

```
function dfs(depth, start) {
  if (depth >= 1) { // 현재 조합에 대하여 결과 계산
    let totalX = 1;
    let totalY = 0;
    for (let i of result) { // 인덱스(index)를 하나씩 확인하며
      let [x, y] = arr[i];
      totalX *= x;
      totalY += y;
    }
    answer = Math.min(answer, Math.abs(totalX - totalY));
  }
  for (let i = start; i < n; i++) { // 모든 조합 계산
    if (visited[i]) continue;
    visited[i] = true; // 방문 처리
    result.push(i);
    dfs(depth + 1, i + 1); // 재귀 함수 호출
    visited[i] = false; // 방문 처리 해제
    result.pop();
  }
}
```

도영이가 만든 맛있는 음식: https://www.acmicpc.net/problem/2961

정답 코드 예시

JavaScript 백트래킹
백트래킹 문제 풀이

```
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

let n = Number(input[0]);
let arr = [];
for (let i = 1; i <= n; i++) { // 각 재료의 (신맛, 쓴맛) 기록
  let [x, y] = input[i].split(' ').map(Number);
  arr.push([x, y]);
}
let visited = new Array(n).fill(false); // 방문 처리 배열
let result = []; // 조합 결과 배열
let answer = 1e9;

dfs(0, 0);
console.log(answer);
```

로또: https://www.acmicpc.net/problem/6603

혼자 힘으로 풀어보기

문제 제목: 로또

문제 난이도: ★★★☆☆

문제 유형: 백트래킹, 경우의 수, 완전 탐색

추천 풀이 시간: 50분

로또: https://www.acmicpc.net/problem/6603

문제 해결 아이디어

- 원소가 K개인 집합 $S = \{1, 2, 3, 5, 8, 13, 21, 34\}$가 있다고 해보자.
- 로또이기 때문에, 이 중에서 6개를 골라야 한다.
- 현재 원소의 개수는 $K = 8$개이므로, 가능한 **모든 조합의 수**는 다음과 같다.
- $Combination(8,6) = 28$

```
[1, 2, 3, 5, 8, 13]
[1, 2, 3, 5, 8, 21]
[1, 2, 3, 5, 8, 34]
[1, 2, 3, 5, 13, 21]
...
[3, 5, 8, 13, 21, 34]
```

로또: https://www.acmicpc.net/problem/6603

JavaScript
백트래킹
백트래킹
문제 풀이

문제 해결 아이디어

- 본 로또 문제의 요구사항은 다음과 같다.
- K는 최대 $6 \leq K \leq 13$이다.
- 이 중에서 6개를 골라야 한다.
- 최대 경우를 생각하면 다음과 같다.
- $Combination(13,6) = 8{,}580$

로또: https://www.acmicpc.net/problem/6603

정답 코드 예시

```
function dfs(arr, depth, start) {
  if (depth == 6) { // 모든 조합을 확인하는 부분(로또는 6개의 자연수로 구성)
    let result = []; // 조합(combination) 결과 저장 테이블
    for (let i of selected) result.push(arr[i]);
    for (let x of result) answer += x + " "; // 계산된 조합을 실질적으로 처리하는 부분
    answer += "\n";
    return;
  }
  // start 지점부터 하나씩 원소 인덱스(index)를 확인하며
  for (let i = start; i < arr.length; i++) {
    if (visited[i]) continue; // [중복을 허용하지 않으므로] 이미 처리 된 원소라면 무시
    selected.push(i); // 현재 원소 선택
    visited[i] = true; // 현재 원소 방문 처리
    dfs(arr, depth + 1, i + 1); // 조합이므로, 재귀 함수 호출시 다음 인덱스(index)를 넣기
    selected.pop(); // 현재 원소 선택 취소
    visited[i] = false; // 현재 원소 방문 처리 취소
  }
}
```

로또: https://www.acmicpc.net/problem/6603

JavaScript 백트래킹
백트래킹 문제 풀이

정답 코드 예시

```
let fs = require('fs');
let input = fs.readFileSync('/dev/stdin').toString().split('\n');

for (let i = 0; i < input.length; i++) {
  let data = input[i].split(" ").map(Number);
  if (data[0] == 0) break; // 테스트 케이스 종료 조건
  else {
    n = data[0];
    arr = data.slice(1);
    visited = new Array(n).fill(false); // 각 원소 인덱스(index)별 방문 여부
    selected = []; // 현재 조합에 포함된 원소(element)의 인덱스
    answer = "";
    dfs(arr, 0, 0);
    console.log(answer);
  }
}
```