

풀이

1년 전쯤 풀었던 문제. 기본적으로 풀이 방법은 동일하다. 이전에 가장 싸게 살 수 있었던 지점을 기억해놓고 그 지점보다 싼 곳이 나오지 않는 이상 그 가격을 기준으로 거리를 계산하면 된다.

자바스크립트로 풀이시 계산 과정에서 `Number.MAX_SAFE_INTEGER` 를 초과할 수 있어서 `BigInt` 형을 사용해야 한다. 또 답안 제출시에도 `toString()` 으로 `BigInt` 에 붙은 `n` 을 제거해줘야 한다.

```
const [n, roads, cities] = require('fs').readFileSync('./data.in').toString().trim().split('\n');
const nodes = cities.split(' ');
const edges = roads.split(' ');

let prev = BigInt(nodes[0]), sum = prev * BigInt(edges[0]);
for (let i = 1; i < edges.length; i += 1) {
  const node = BigInt(nodes[i]), edge = BigInt(edges[i]);
  if (node < prev) prev = node;
  sum += edge * prev;
}

console.log(sum.toString());
```

이전 풀이.

```
const sc = require('fs').readFileSync('/dev/stdin').toString().trim().split('\n');
const cities = parseInt(sc[0], 10);
const [edges, nodes] = sc.slice(1).map(v => v.split(' ').map(e => parseInt(e, 10)))
let min = BigInt(nodes[0]);
let cost = 0n;
for (let i = 0; i < cities - 1; i += 1) {
  if (nodes[i] < min) {
    min = BigInt(nodes[i]);
  }
  cost += min * BigInt(edges[i]);
}
console.log(`${cost}`);
```