

## LIS (Longest Increasing Sequence)

0, 8, 4, 12, 2, 10, 6, 14, 1, 9, 5, 13, 3, 11, 7, 15

LIS(Longest Increasing Subsequence)란 주어진 배열에서 가장 길게 증가하는 부분 수열을 찾는 문제입니다. 즉, 주어진 배열에서 임의의 숫자  $i$ 에서  $j$ 로 갈 수 있을 때  $i < j$ 인 경우에만 갈 수 있다면, 이때 가장 길게 이어지는 부분 수열을 찾는 것입니다.

### LIS ( with Javascript )

```
function lis(arr) {
  let dp = new Array(arr.length).fill(1);
  let max = 0;

  for (let i = 1; i < arr.length; i++) {
    for (let j = 0; j < i; j++) {
      if (arr[i] > arr[j] && dp[i] < dp[j] + 1) {
        dp[i] = dp[j] + 1;
      }
    }
  }

  for (let i = 0; i < dp.length; i++) {
    if (max < dp[i]) {
      max = dp[i];
    }
  }

  return max;
}

let arr = [3, 2, 6, 4, 5, 1];
console.log(lis(arr)); // 3 (최장 부분 증가 수열: [2, 4, 5] 또는 [2, 4, 6])

// arr:  3 2 6 4 5 1
// dp:   1 1 2 2 3 1
```

위 코드에서는 DP 배열 `dp` 을 생성하여 각 원소를 마지막에 넣을 경우의 가장 긴 부분 수열을 저장합니다. 이때, `dp[i]` 는 `arr[i]` 를 마지막으로 하는 부분 수열의 길이를 나타냅니다. 그리고 배열 `dp` 를 모두 탐색하여 가장 큰 값을 반환하면 됩니다.

따라서, 가장 긴 증가하는 부분 수열의 길이는 3 입니다.