Используем жадный алгоритм.

function maxActivities(activities) {

// Сначала сортируем активности по времени окончания

activities.sort((a, b) => a[1] - b[1]);

// Инициализируем массив для выбранных активностей

const selectedActivities = [];

// Первая активность всегда выбирается, так как она имеет самое раннее окончание

selectedActivities.push(0);

// Последнее время окончания выбранной активности

let lastEndTime = activities[0][1];

// Просматриваем остальные активности

for (let i = 1; i < activities.length; i++) {

if (activities[i][0] >= lastEndTime) {

// Если начало следующей активности больше или равно времени окончания предыдущей,

// то эта активность может быть выбрана

selectedActivities.push(i);

lastEndTime = activities[i][1];

}

}

return selectedActivities;

}

// Пример

const activities = [[1, 4], [3, 5], [0, 6], [5, 7], [3, 8], [5, 9], [6, 10], [8, 11]];

const result = maxActivities(activities);

console.log(result); // Результат: [0, 3, 7]

Этот код сначала сортирует активности по времени окончания, а затем выбирает активности, которые не перекрываются с предыдущими. В результате, вы получите массив индексов выбранных активностей, которые максимизируют общее количество активностей без перекрытия.