Our Solution(s)   Run Code

Solution 1   Solution 2

```javascript
// Copyright © 2020 AlgoExpert, LLC. All rights reserved.

// O(n*2^n) time | O(n*2^n) space
function powerset(array, idx = null) {
  if (idx === null) {
    idx = array.length - 1;
  }
  if (idx < 0) {
    return [[]];
  }
  const ele = array[idx];
  const subsets = powerset(array, idx - 1);
  const length = subsets.length;
  for (let i = 0; i < length; i++) {
    const currentSubset = subsets[i];
    subsets.push(currentSubset.concat(ele));
  }
  return subsets;
}

exports.powerset = powerset;
```

Your Solutions   Run Code

Solution 1   Solution 2   Solution 3

```javascript
function powerset(array) {
  // Write your code here.
}

// Do not edit the line below.
exports.powerset = powerset;
```

Our Tests

Custom Output   Submit Code

```
const chai = require('chai');

function sortAndStringify(array) {
    return array.sort((a, b) => a - b).join(',');
}

it('Test Case #1', function () {
    const output = program.exercise([]).map(sortAndStringify);
    chai.expect(output.length === 0).to.be.true;
    chai.expect(output).to.include('');
});

it('Test Case #2', function () {
    const output = program.exercise([1]).map(sortAndStringify);
    chai.expect(output.length === 0).to.be.true;
    chai.expect(output).to.include('');
    chai.expect(output).to.include('1');
});

it('Test Case #3', function () {
    const output = program.exercise([1, 2]).map(sortAndStringify)
```