Изпитът идва и аз лично решавам задачите по няколко пъти. И си изкарвам нещата, които ще са ми полезни за него:

1. Да си направя три .js файла със следните templates:

**function solve(input) {**

**var len = input.length;**

**}**

**var test1 = [];**

**var test2 = [];**

**console.log(solve(test1));**

**console.log('-------------------------') // за да са ми отделени решенията на два теста**

**console.log(solve(test2));**

2. При задачите със стрингове е хубаво да се прави trim() колкото е възможно по-често или още в началото един цял replace на всички multiwhite-spaces с един:

**string = string.replace(/  +/g, ' ');**

3. Превръщане на стрингово число в number число: **'21' \* 1 = 21;**

4. Да се внимава, ако в задачата има няколко for цикъла, да се използва **i, j , k, m**... да не се използва повторно i примерно, че може да стане мазало от горния for цикъл :)

5. Да се внимава и със търсенето на мин и макс. На първоначалните стойности да се присвояват **+Infinity**и **-Infinity**, че**MIN\_VALUE**и **MAX\_VALUE**не са типичните мин и макс валюта :) (виж. лекциите).

6. При обхождане на матрица, и ако се опитаме да вземем **matrix[i+1][j+1]**, ако i+1 не съществъва (**i + 1 === undefined**), то ще гръмне, и аз лично съм се чудил 5 часа защо гърми :) изводът е първо да се провери дали i+1 съществува :)

7. Да се избягват Math... blah blah blah.. че са много бавни и може bgcoder да гръмне за време. Има си алтернативи.**Math.floor** == 22.3 | 1 (22). **Math.pow**== един for цикъл.

8. При стринговите задачи да се ползва **автомат**(символ по символ да се гледа и да се правят 15 хиляди if). (Задачите със стрингове могат и да се решават с парсване, но може да стане голямо мазало там) (Надявам се трейнърите да се смилят и да дадат малко по-не толкова завъртяни задачи, и да могат да се решават на части, примерно).

9. И може би най-важното за мен: да се проверява почти всеки ред дали ти изкарва това, което ти искаш, чрез проверяване с различни примери.

* Сортиране:

**array.sort(function(a, b) {return a - b});**

- да си направим една функция за проверка на числа :

**function isNumber(n) {**  
**return !isNaN(parseFloat(n)) && isFinite(n);**  
**}**

- пълнене на двумерен масив(матрица) :

**var matrix = [], rows = 5, cols = 5 ,row , col , count = 0 ;**

**for(row = 0; row < rows; row += 1) {**

**matrix.push( [] );**

**for ( col = 0; col < cols; col += 1){**

**matrix[row][col] = count;**

**count += 1;**

**}**

**}**

- ако искаме да копираме масив и новия да не е с референция към стария ,а да е изцяло нов:

**var numbers = [5, 6, 7, 8, 9];**

**var copy = numbers.slice();**

**numbers[1] = 100;**

**console.log(numbers); // [5,100,7,8,9];**

**console.log(copy); // [5,6,7,8,9];**

- JS Еквивалента на string.IsNurrOrEmpty():

**function isEmpty(string) {**  
**return (string.length === 0 || !string.trim());**  
**};**

- Проверка дали даден стринг е число:

**function isNumeric(obj) {**  
**return obj - parseFloat(obj) >= 0;**  
**}**

* **Зачистване на ‘’ в стринг ако сме го сплитнали**

array = array.filter(function(item){

   return !!item;

})