

Array Test Memory Work – Speeds & sorting

1. Speed of swap	$O(1)$
2. Speed of find length of array	$O(1)$
3. Speed of max	$O(n)$
4. Speed of print	$O(n)$
5. Speed of min	$O(n)$
6. Speed of average	$O(n)$
7. Speed of search	$O(n)$
8. Speed of bin sort	$O(n)$
9. Speed of selection sort	$O(n^2)$
10. Speed of bubble sort – best case	Close to $O(n)$
11. Speed of bubble sort – average case	$O(n^2)$
12. Speed of bogosort	$O(n!)$
13. In Big-Oh notation, the O stands for...	Order
14. In Big-Oh notation, the n stands for..	Array size
15. Why don't we measure algorithm speed in terms of seconds or time?	It is hardware dependent.
16. Which is faster: bubblesort (average) or selection sort?	Selection Sort
17. Put the 3 sorts in order, fastest to slowest	Bin sort Bubble (best case), Selection, Bubble (average case)
18. Why is selection sort called selection sort?	SELECTING best element (max)
19. Why is bubblesort called bubblesort?	The billions of little swaps are like BUBBLES rising in pop.
20. Why is bin sort called bin sort?	It makes BINS to count amount of each element
21. Term for putting elements in order	Sorting
22. Good hardware cannot compensate for	A bad algorithm

23. Term for a series of steps that complete a task	Algorithm
24. Three tests to determine if something is an algorithm.	1. series of steps 2. for a (one) task 3 completes the task
25. Is all computer code an algorithm?	Yes
26. Are all flowcharts an algorithm?	Yes
27. Which sorting algorithm is not “in-place”?	binsort
28. What does “in-place” algorithm mean?	It uses swaps. It doesn't need extra memory.
29. What is the slowest sorting algorithm?	Bogosort
30. Who wrote that good hardware cannot compensate for a slow algorithm?	Jon Bentley
31. What is the first test to determine which sorting algorithm to use?	Repeated integers that fall in a small range. Use bin.
32. What is the second test to determine which sorting algorithm to use?	Almost sorted. Use Bubble.
33. Put the sorting speeds in order, fastest to slowest	$O(1)$, $O(\log n)$, $O(n)$, $O(n \log n)$, $O(n^2)$, $O(n^3)$, $O(n^4)$, $O(n!)$
34. What is a compromise when you must give up one thing to gain another?	A trade-off.
35. Does one best sorting algorithm exist?	No. You must make trade-offs when selecting which to use.
36. Positive of integers for picture names	Easy to type (no "" or .equals or .compareTo)
37. Negative of integers for picture names	Not meaningful.
38. Positive of Bubble sort	If almost sorted, close to $O(n)$. That's fast
39. Negative of Bubble sort	In all other cases, slow $O(n^2)$. A lot of swaps.

40. Positive of Selection sort	Easy to understand. Based on max.
41. Negative of Selection sort	Slow. Simplicity isn't efficient.
42. Positive of Bin sort	Fast. Really fast. $O(n)$
43. Negative of Bin sort	Speed comes from very narrow situation: repeated ints, in a small range.
44. What is the name for the address of the array elements?	index
45. What is the plural of index?	indices
46. The length of the array is 10. What are the indices?	0-9
47. The indices of an array are 0-5. What is the length?	6
48. What is the name for the data located at each index?	element
49. What is the name for a set of variables all grouped under one name?	array
50. 3 Reasons arrays are useful.	<ol style="list-style-type: none"> 1. Loop through them to reduce lines of code 2. Sorting 3. Searching
51. 3 Reasons sorting is useful	<ol style="list-style-type: none"> 1. Searching is easier 2. Finding min, max is easier 3. Finding duplicates is easier
52. How do you convert from a 1D array to a 2D array?	Add a set of []
53. How many steps to swap two elements?	3 steps
54. What is an array index out of bounds error?	Calling a place off the end of the array

55. If you are in [x][y], what is the co-ordinates for one spot above you?	[x-1][y]
56. If you are in [x][y], what is the co-ordinates for one spot below you?	[x+1][y]
57. If you are in [x][y], what is the co-ordinates for one spot to the left?	[x][y-1]
58. If you are in [x][y], what is the co-ordinates for one spot to the right?	[x][y+1]
59. Which way is the row?	Horizontal (i)
60. Which way is the column?	Vertical (j)
61. Which way is x or i?	Horizontal (Row)
62. Which way is y or j?	Vertical (Column)
63. What is the edge guard for i-1?	i-1 >=0
64. What is the edge guard for i+1?	i+1 < row
65. What is the edge guard for j-1?	j-1 >=0
66. What is the edge guard for j+1?	j+1 < col
67. What are the steps to move a maze character?	<ol style="list-style-type: none"> 1. Check if new spot is OK 2. Erase old spot 3. Adjust x, y co-ordinates 4. Draw in new spot
68. How many loops do you need for a 1D coding question?	1
69. How many loops do you need for a 2D coding question?	2
70. What is the loop that occurs frequently in 1D algorithms?	for(int i=0; i<a.length, i++)
71. What is the outer for loop for a 2D coding question?	for(int i=0; i<row; i++)
72. What is the inner for loop for a 2D coding question?	for(int j=0; j<col; j++)