

Sorting Homework

Due Monday, March 3, 2008 at 9:00am in class

There are four required problems for this assignment: problems from chapter 6, pages 145-146: **1, 6, 8**, and the problem labeled S below. See the hint on the back of this page.

S. Sort the following integers by their **absolute value** (i.e., Math.abs). For example, if you were to sort only the numbers {5, -1, 0, 2}, the correct answer would be {0, -1, 2, 5}. These integers were generated by calling the generateNumbers method on the course web page with the arguments *max* = 100 and *n* = 800. You may use any tools that you like to solve this problem, including your solution to this week's lab. Please print your answer in a 20 x 80 grid like this so that it is easy for me to check your solutions.

99	79	-91	-55	-20	18	-24	18	-6	48	79	90	76	-80	-76	55	3	-81	31	42
92	85	-60	20	13	-95	-9	-26	82	-33	12	0	9	-97	-85	-19	-72	-23	-20	-10
-32	-27	82	-69	20	36	-33	-61	74	-97	-98	43	32	76	35	23	93	-23	1	
33	95	-88	-46	79	30	-14	43	0	-53	-5	-49	55	-79	28	-47	0	91	63	-78
-53	5	7	-41	26	-23	12	-3	48	-82	-19	-69	-41	18	-47	48	-36	-71	-2	81
-68	13	93	-42	-96	-60	17	14	92	56	22	68	-35	92	46	-66	-30	12	15	-76
36	-18	-88	-71	-50	-99	-36	-41	24	72	41	-41	83	-59	99	-78	26	-85	28	50
99	48	-13	3	6	-93	-33	82	-6	80	83	59	-95	20	60	-32	50	52	67	27
-77	-34	-81	-86	-3	-80	-54	20	-23	-3	26	-97	-59	-28	64	10	-13	20	-69	31
-95	45	-61	74	10	0	24	-37	-74	35	-51	-86	-34	-84	-15	-39	-91	55	91	-86
-50	41	19	-72	64	-19	-44	65	67	-27	-88	-21	32	-81	41	-72	-87	61	-44	-60
35	96	32	-56	62	-68	-97	-45	9	68	-46	99	42	34	-73	-46	-2	-52	-70	71
-62	-89	33	8	24	-99	75	45	17	7	49	1	50	-36	-38	-34	23	-11	-18	63
10	29	54	-75	-45	-69	-79	56	-66	-70	-83	-9	-67	-94	-40	69	-79	38	-42	6
-77	44	-54	-48	40	25	-15	55	70	99	85	26	-92	30	39	-97	65	-9	-83	-54
-20	-66	-91	13	61	-62	10	93	84	-39	39	-83	45	-23	72	-48	56	-57	8	44
-41	-28	-22	-38	-70	-88	56	38	-84	-66	-55	50	35	51	-36	51	94	87	-89	88
-70	59	93	80	64	-15	26	59	-56	-31	7	30	-44	-47	86	-69	50	6	80	57
-3	88	46	-99	86	60	3	78	-82	62	38	66	-38	62	-5	-49	-65	-60	94	13
-68	-93	-21	-4	-73	91	-63	-60	-99	16	16	85	-39	46	79	-57	6	20	-90	28
53	15	47	-80	10	-73	15	-48	-84	-13	5	95	-25	73	96	-74	45	-67	79	-37
70	7	68	67	8	67	-57	62	-38	10	-35	59	-77	57	-26	-16	-48	-13	-12	-74
75	4	3	31	-20	-60	51	54	58	62	-97	-7	-99	-76	36	47	-32	80	-39	-80
-61	58	-86	-10	-70	73	4	-97	-39	67	-87	-90	79	70	12	-16	-89	-3	36	47
80	-42	50	-5	-40	-68	-52	-42	17	49	93	21	77	-54	-67	-68	72	-21	-4	-40
30	-63	-15	-32	5	-4	-32	-67	-48	-52	11	-39	35	-57	6	6	-70	-24	66	93
30	-98	-51	-88	-29	-61	-75	90	61	-7	31	-47	-32	-45	27	-76	99	0	-79	30
-89	-15	27	-85	52	-67	72	13	-11	92	12	-64	-8	-95	90	-16	-57	54	-40	24
43	-31	95	20	-82	28	-4	-42	-19	60	73	-66	-15	16	-98	-16	-74	-52	-42	-92
99	61	-93	-32	22	59	51	72	-4	88	2	-9	-65	-74	29	-25	0	82	-21	-54
-59	95	64	53	-44	-68	-82	-63	0	-91	64	60	-21	-16	-31	-6	-50	61	19	-54
-40	28	-51	-91	30	20	79	7	-38	-6	89	54	76	54	54	30	33	-56	-17	-98
-87	39	-8	-87	-30	81	46	8	-60	14	89	-72	53	43	-76	-17	24	40	-54	52
72	92	-60	46	-81	24	-52	-11	-25	-85	-94	-1	17	-40	-68	-76	36	-12	47	70
96	-80	-57	-20	-22	-47	-47	48	83	-93	-26	-84	54	20	41	72	-87	90	98	-68
32	-11	-73	-56	-4	28	84	20	-7	26	16	94	-7	16	37	-96	-67	-71	55	-69
-89	-86	53	-46	-14	8	0	-56	-34	7	-31	63	-59	96	45	-92	11	3	-70	-22
48	34	-78	-44	80	40	-23	12	-70	95	11	-84	-7	53	-46	59	36	-94	-15	-98
-70	-39	40	-32	99	-16	25	-75	-75	43	-36	45	-71	1	-66	91	75	95	-50	62
75	-80	-98	-82	6	-84	-36	0	59	42	-70	-23	-7	76	-63	9	20	-48	-72	21

Tip for question #8: when a method calls another method (or itself recursively), some space must be consumed in remembering the information about the caller. When you evaluate a recursive method by hand, that space is your “TODO list” of problems to get back to. On a computer it is called a “stack frame” for reasons we’ll see later in the semester. In Java, each stack frame requires constant space. These stack frames count towards the total space consumption of a method.

The following problems are optional. Thinking about them may help you prepare for the exam. If you complete them I will mark and give feedback on your answers, but you will not gain or lose any points on this assignment based on your answers.

Consider the following method:

```
public static String funny (int n) {  
    int x = 1 << n;  
    long y = 0; // long is like int, but has 64 bits instead of 32  
    String s = “”;  
    for (int i = 0; i < x; ++i) {  
        s = “” + i;  
        y = y * 1103515245 + 12345;  
    }  
    return s + “:” + y;  
}
```

- A. Method funny returns a String containing two numbers separated by a colon. What is the *first* of the two numbers in the string that funny(6) returns?
- B. What is a simple change that will improve the efficiency of this method by a constant factor?
- C. Rewrite method funny so that it does not use FOR or WHILE loops but still performs essentially the same function (you will need a helper method).
- D. Derive a tight upper bound on the run time of method funny, as it is written above.
- E. Derive a tight lower bound on the run time of method funny, as it is written above.
- F. Derive a tight upper bound on the space required by method funny, as it is written above. *This is tricky, in part because it requires you to speculate about aspects of the Java language that we have never discussed; your ideas are more important than the answer.*