HW 2, Part II

Due 23:00 October 12, 2017

CS ID:

Section (circle one):

Monday	L1J 9–11a	L1M 11a–1p	L1C 4–6p
Tuesday	L1A 11a–1p	L1F 4–6p	
Wednesday	L1B 9–11a	L1E 2–4p	L1M 6–8p
Thursday	L1D 10:30a–12:30p	L1G 4–6p	
Friday	L1K 1–3p	L1H 3–5p	

Please print out this document and write your solutions into the spaces provided. Show your work where necessary for full credit. If you require additional space, please indicate in the question space that you are writing on the last blank page, and also indicate on the blank page which question the work solves.

You must scan and upload the completed document, including this page and the last page, to GradeScope, using your 4- or 5-character CS ID.

1. **[Divvy Sort – 22 points].**

   In class we discussed 2 different iterative sorting algorithms: selection and insertion. Here is code for a third, which we're calling "Divvy Sort". (The `swap` function simply switches the values of the two parameters.)

```
void DivvySort(vector<string> & a)
{
     int n = a.size();
     int count = 0;
     for(int i = n/2; i > 0; i = i/2) {
         for(int j = i; j < n; j++) {
            for(int k = j; (k-i) >= 0; k = k-i) {
                 if (a[k] < a[k-i]){
                     swap(a[k],a[k-i]);
                 }
             }
         }
     }
 }
```

   (a) (4 points) You will use the table below to trace the execution of this algorithm on a small example. Simply enter any four integers into the array in iteration 0, and go!

   - Be very careful to increment (and decrement) the indices accurately.
   - Take a moment to look at the state and structure of the data after every iteration.

   (b) (2 points) The inner two loops of "Divvy Sort" are very similar to an algorithm you have seen before.

   Write its name in the box:

   (c) (2 points) Navigate to the web app found here: https://www.toptal.com/developers/sorting-algorithms, and determine which of the algorithms is equivalent to "Divvy Sort."

   Write its name in the box:

   (d) (2 points) Wikipedia gives a table of running times that apply to a variety of adaptations to the algorithm. Use that table to figure out the bound on the running time of *our* implementation (above).

   Write the asymptotic running time in the box:

   (e) (4 points) Express the running time of "Divvy Sort" as a sum:

$$\sum_{i=\_\_\_\_}^{\overline{\qquad}} \sum_{j=\_\_\_\_}^{\overline{\qquad}} \sum_{k=\_\_\_\_}^{\overline{\qquad}} \underline{\hspace{3cm}}$$

iteration	array				i	j	k
0							
1							
2							
3							
4							
5							
6							
7							
8							
9							
10							
11							
12							
13							

(f) (4 points) Solve the sum to prove the asymptotic running time that you found on wikipedia. To do this, you will use your answer from part (b), together with the fact that $O(f(n) + g(n)) = O(f(n)) + O(g(n))$. Show your work!

(g) (4 points) Look again at the web app from part (c). Explain why "Divvy Sort" finishes significantly faster than your answer in part (b).

2. [**Sorting Mechanics – 24 points**]. In this problem you will show us that you understand the concept of *loop invariants* in sorting algorithms. Each of the lists of names below has been created by invoking a sorting algorithm, and stopping it after some number of iterations, k. For each list, and for each sorting algorithm, fill in the box with the *maximum* possible value of k. In the table, "Sel" stands for selection sort, "Ins" stands for insertion sort, and "Div" stands for divvy sort. In this problem we are considering an "iteration" to be one execution of the *outer* loop of the algorithm.

#	ex 1	ex 2	ex 3	ex 4
0	Asuna	Asuna	Akame	Erza
1	Winry	Akame	Akeno	Jubia
2	Shiro	Akeno	Asuna	Lucy
3	Yuri	Lucy	Erza	Lucy
4	Lucy	Riza	Jubia	Maka
5	Shana	Jubia	Lucy	Nami
6	Erza	Erza	Taiga	Riza
7	Saeko	Nami	Yuri	Shana
8	Taiga	Saeko	Shana	Shiro
9	Jubia	Maka	Nami	Taiga
10	Lucy	Lucy	Shiro	Winry
11	Maka	Taiga	Saeko	Yuri
12	Nami	Shiro	Maka	Asuna
13	Akame	Shana	Lucy	Akame
14	Akeno	Yuri	Riza	Akeno
15	Riza	Winry	Winry	Saeko

Div	□	Div	□	Div	□	Div	□
Ins	□	Ins	□	Ins	□	Ins	□
Sel	□	Sel	□	Sel	□	Sel	□

Blank sheet for extra work.