Coursework 2. Submission deadline 20/12/20

November 20, 2020

For all the questions where it is required to design an agorithm subject to a runtime constraint, please provide an explanation as to why you believe your algorithm satisfies this contraints. Solutions without explanation will not be marked.

1 Linear time testing for restricted lists (20 marks)

Let A be a list all elements of which are in the range $0, \ldots, len(A) - 1$. Let B be an arbitrary list. Design an $O(n)$ algorithm that returns $true$ if A contains an element that is not contained in B. Otherwise the algorithm should return $false$. Hint: consider the use of list Ind as in the end of Section 2.5. ofthe Runtime notes.

2 Finding a small number of distinct elements (20 marks)

Each of the following two problems is worth 10 marks.

1. Design an $O(\log n)$ algorithm whose input is a sorted list A. The algorithm should return $true$ if A contains at least 3 distinct elements. Otheriwse the algorithm should return $false$.
2. Design an $O(\log n)$ algorithm whose input is a sorted list A. The algorithm should return $true$ if A contains at least 4 distinct elements. Otheriwse the algorithm should return $false$. Hint: run the function for part 1 at most three times.

3 Processing a sorted list (2o marks)

Let A be a sorted list. Solve the following exercises whose input in A.

1. **5 marks.** Write an $O(n)$ function that returns an element of A having the largest number of occurrences in A. For example, if $A = [1, 2, 2, 8, 8, 8, 10, 10, 15, 15, 15, 20]$ the algorithm should return either 8 or 15.

2. **5 marks.** Write an $O(n)$ function that returns a sorted list B with the same elements as in A but without multiple occurrences. For example, the output for the list A as in the previous part is $[1, 2, 8, 10, 15, 20]$.

3. **10 marks.** Write a function that returns a list C that is a permutation of the elements of B so that they are sorted according to the following cirterion. $B[i]$ occurs C before $B[j]$ if

   - The number of occurrences of $B[i]$ in A is greater than the number of occurrences of $B[j]$ in A or
   - The number of occurences of $B[i]$ in A is the same as that of $B[j]$ but $B[i] < B[j]$. For lists A and B as in the examples in the previous two parts, $C = [8, 15, 2, 10, 1, 20]$.

4 Individual statistics of matrix elements (20 marks).

For the following three tasks the input is a matrix A. As a running example, we consider the following matrix.

```
7  2  1
9  2  5
7  5  2
```

1. **5 marks.** Write an algorithm that returns a list B containing all the number occurring in A without repetitions. The numbers can appear in B in an arbitrary order. For the example,the output can be $[7, 2, 1, 9, 5]$.

2. **10 marks.** Write an algorithm that returns a list C that is a permutation of the elements of B so that they are sorted according to the following cirterion. $B[i]$ occurs C before $B[j]$ if

   - The number of *rows* in A containing an occurrence of $B[i]$ is greater than the number of *rows* in A containing an occurrence of $B[j]$ or
   - The number of rows in A containing an occurrence of $B[i]$ is the same as that of $B[j]$ but $B[i] < B[j]$. For lists A and B as in the examples in the previous two parts, $C = [2, 5, 7, 1, 9]$.

3. **5 marks.** Write an algorithm that returns a list C that is a permutation of the elements of B so that they are sorted according to the following cirterion. $B[i]$ occurs C before $B[j]$ if

   - The number of *columns* in A containing an occurrence of $B[i]$ is greater than the number of *columns* in A containing an occurrence of $B[j]$ or
   - The number of columns in A containing an occurrence of $B[i]$ is the same as that of $B[j]$ but $B[i] < B[j]$. For lists A and B as in the examples in the previous two parts, $C = [2, 5, 1, 7, 9]$.

5 Merging and sort the rows of a matrix

Write a function whose input is a square matrix A whose rows are sorted lists. The function should return a list B obtained from A by mergining all the rows into a single list and then sorting the resulting list. For example for A equal

```
1 2 8
2 3 9
3 4 4
```

The resulting list is $[12, 2, 3, 3, 4, 4, 8, 9]$. The runtime of the algorithm should be $O(n^2 \log n)$. Hint: use Mergesort.