CSCI-B 405, Applied Algorithms - Assignment 1

Due by Thursday 9/19/2024
Total points 100
* Submitted pseudo-code should be typed digitally using any software tool or format, such as MS Word, or Latex.

1 Stable Matching

Points: 10% Explain the best-case scenario in Gale-Shapley's stable matching algorithm within the scope of running time, then give an estimation of the running time in the best-case. You can derive your answer from the two genders matching example.

Answer: The best-case scenaio the stable matching algorithm is going to be O(n*k) where k is a smaller constant compared to n. This is because in the best case we find a match on the first try every single time, rather than having to check the match each and every time.

2 Loop Invariants

Points: 15% Write pseudo-code for an algorithm whose input is a list of n positive integers and whose output is (ℓ, h), where ℓ is the smallest value in the list and h is the largest. Prove your algorithm is correct using a loop invariant.

PseudoCode:

Algorithm FindMinMax(List): Input: List of positive integers of size n Output: Tuple (l, h) where l is the smallest and h is the largest value in the list

if n == 0: return (undefined, undefined) // Handle edge case of empty list

l = List[0] // Initialize l to the first element h = List[0] // Initialize h to the first element

for i from 1 to n-1: if List[i] ¡ l: l = List[i] // Update l if the current element is smaller else if List[i] ¿ h: h = List[i] // Update h if the current element is larger

return (l, h) // Return the smallest and largest values

proof pf: Initiailization: Before the loop starts, when i = 1, we initialize l and h to List[0]. l and h correctly represent the minimum and maximum values of the sublist List[0:1], which only contains the first element of the list. Thus the invariant trivially holds before the first iterization. Maintenance: During the loop: For each index i from 1 to n-1, we perform the following checks: If List[i] ¡ l, we update l to List[i]. After this update, l still correctly represents the minimum value of the sublist List[0:i] If List[i] ¿ h, we update h to List[i]. After this update, h still represents the maximum value of teh sublist List[0:i] After processing element List[i], l, and h reflect the min and max val's of the sublist List[0,i+1] Termination When the loop terminates (i = n), the loop invariant ensures that l and h represent min and max values of the entire list. Thus, the final values of l and h are the smallest and largest values in the list.

3 Descending Sorting

Points: 15% A - Write pseudo-code for an algorithm that sorts an array in descending order by iterating over the items. In the first iteration the algorithm finds the largest value, then add it to the output array. In the following iteration, the algorithm finds the next largest value and add it to the output array, and so on. Provide extensive comments to explain the steps for full credit.

Points: 15% B - Estimate the worst-case running time of the algorithm and provide a justification. Your justification can be of English words only or algorithm lines analysis with notations.

4 Sorting Algorithms Experimental Study

Points: 35% Write a program to report the running time (in microseconds) of the insertion sort algorithm and the merge sort algorithm for the three different scenarios, that are: the best-case scenario (numbers already sorted), the worst-case scenario (numbers reversed), and the average-case scenario (numbers are randomized). The three scenarios should run for three different array lengths, that are: 100, 1000, and 10,000. Meaning that your program should report 18 experiment. The program output should provide sufficient labels to understand the results and comments.

For example, the results when running an experiment on a 1000-items array could look like this:

```
insertion sort best-case xxx microseconds
insertion sort worst-case xxx microseconds
insertion sort average-case xxx microseconds
merge sort best-case xxx microseconds
merge sort worst-case xxx microseconds
merge sort average-case xxx microseconds
```

- The language to be used for this part can be (Python, Java, JavaScript, C#, C/C++)
- The program code should be submitted separately in a single file (no screenshots)

Points: 10% Explain your experimental observations using English words, statistically in tables, or visually using charts.