

Problem Set 0 Solutions

Released 24 Jan 2024

Due 31 Jan 2024 10:30pm in Gradescope

Instructions. You may discuss the material with your classmates, but you must write solutions yourself. List collaborators on your submission. Do not use other sources of help other than the course staff and the textbook.

If you are asked to design an algorithm, please provide: (a) the pseudocode or precise description in words of the algorithm, (b) an explanation of the intuition for the algorithm, (c) a proof of correctness, (d) the running time of your algorithm and (e) justification for your running time analysis.

Submissions. Please submit a PDF file. You may submit a scanned handwritten document, but a typed submission is preferred. Please assign pages to questions in Gradescope. By Homework 2 typed submissions will be required

1. Termination of Gale-Shapely Prove that the Gale-Shapely algorithm terminates in a stable perfect matching.

Solution:

Perfect matching: At the end, there cannot be an applicant and employer both unmatched. An employer left unmatched at the end of the process must have made an offer to all applicants. But an applicant who receives an offer remains employed for the rest of the process, so there can be no unemployed applicants. Since the numbers of applicants and job openings are equal, there can also be no open positions remaining.

Stable match: No applicant X and employer Y can prefer each other over their final match. If Y makes an offer to X , then X would only reject Y after receiving an even better offer, so X cannot prefer Y to their final match. And if Y stops making offers before reaching X in their preference list, Y cannot prefer X to their final match. In either case, X and Y do not form an unstable pair.

Solution from Wikipedia

2. CLRS-2.1-2 Consider the procedure Sum-Array. State a loop invariant for the procedure and show it returns the sum of the numbers $A[1:n]$

Solution: Initialization: Before the loop, $\text{sum}=0$ which is the sum of the first 0 elements of A ($i=0$).

Maintenance: In each iteration of the loop, the value $A[i]$ is added to the prior sum. Sum then consists of the sum of the values in the subarray of $A[1:i]$. Incrementing i thus preserves the loop invariant.

Termination: When $i=n+1$ the loop terminates. By the loop invariant we thus have the sum of $A[1:n]$.

3. CLRS-2.1-3 Rewrite the Insertion-Sort procedure to sort into monotonically decreasing order.

Solution: The only necessary changes are in the while loop, invert the sign from $>$ to $<$ in the phrase $A[j] > \text{key}$. The runtime analysis is unchanged so $\Theta(n^2)$. The only change in the loop

invariant is that it is sorted in monotonically decreasing instead of increasing order.

4. CLRS-2.2-1 Express the function $n^3/1000 + 100n^2 - 100n + 3$ in Θ notation.

Solution: $\Theta(n^3)$

5. CLRS-2.3-4 Use mathematical induction to show that when $n \geq 2$ is an exact power of 2, the solution of the recurrence

$$T(n) = \begin{cases} 2 & n = 2, \\ 2T(n/2) + n & n > 2 \end{cases}$$

is $T(n) = n \log n$ (where $\log$ is $\log_2$)

Solution:

Base Case: when $n = 2$, $T(n) = 2 = 2 \log 2 = 2 * 1 = 2$.

Induction: Let n be a power of 2 greater than 2. Assume $T(n) = n \log n$. Then $2*n$ is the next power of 2. $T(2n) = 2T(n) + 2n = 2n \log n + 2n = 2(n \log n + n) = 2n(\log n + 1) = 2n(\log 2n)$. Thus when $n \geq 2$ is an exact power of 2, $T(n) = n \log n$