

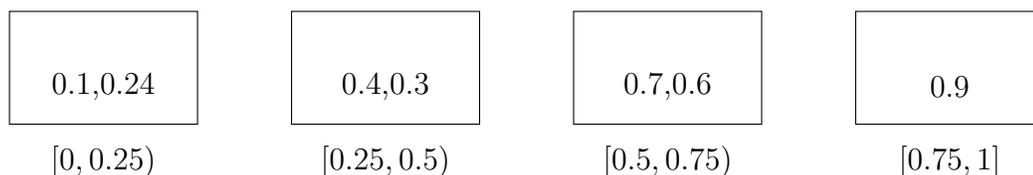
1. We consider the Horner's method for evaluating the polynomial $a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0$ at $x = c$.

```

1 procedure Horner( $c, a_0, a_1, \dots, a_n$ )
2    $y \leftarrow a_n$ 
3   for  $i = 1, 2, \dots, n$ 
4      $y \leftarrow y * c + a_{n-i}$ 

```

- (a) (10 points) Evaluate $3x^2 + 2x + 1$ at $x = 3$ by working through each step of the algorithm showing the values assigned at each assignment step.
- (b) (10 points) Exactly how many multiplications and additions are used by this algorithm to evaluate a polynomial of degree n at $x = c$? (Do not count additions used to increment the loop variable.)
2. Use the Master Theorem to solve the following recurrence relations.
- (a) (5 points) $T(n) = 8T(n/2) + n$
- (b) (5 points) $T(n) = 8T(n/2) + n^3$
- (c) (5 points) $T(n) = 3T(n/2) + n$
- (d) (5 points) $T(n) = T(n/4) + 1$
- (e) (5 points) $T(n) = 3T(n/3) + n^2$
3. (15 points) Consider the input sequence $A = \langle 5, 3, 17, 10, 84, 19, 6, 22, 9 \rangle$. Illustrate the operations of build-heap algorithm on A by specifying the content of A after each step of the algorithm.
4. (15 points) Write an efficient *heapify* algorithm that uses iterative control instead of recursion.
5. (25 points) Instead of a complete sort, the binning problem assigns each element of an input sequence A into bins with specified range. For instance, let $A = \{0.4, 0.7, 0.1, 0.24, 0.3, 0.6, 0.9\}$, and there are four bins as shown below, the binning process will assign 0.1, 0.24 to the first bin, 0.4, 0.3 to the second bin, 0.7, 0.6 to the third bin, and 0.9 to the fourth bin.



Suppose that we are given an input sequence A of n elements a_1, a_2, \dots, a_n within the range $[0, 1]$ and we need to do the binning process using the above four bins. Write an algorithm with linear complexity that does the work.

[*Hint: A decimal number within $[0, 1]$ is presented as a 0–1 string on the computer. For instance, $0.25 = (0.010000000\dots)_2$. If you just look at the substring $(01)_2$, what is the result in base 10?*]

6. **(Bonus)** Consider the adaptive version of the binning problem. In this case, each bin can have no more than s elements, where s is a prescribed parameter. Should that happen, the bin is partitioned into two equal parts. For instance, assume that $s = 3$, and the bin $[0.25, 0.5)$ have four elements 0.27, 0.33, 0.35, and 0.49, the adaptive binning process will partition it into two bins $[0.25, 0.375)$ with 0.27, 0.33, and 0.35 and $[0.375, 0.5)$ with 0.49. Design an algorithm that does the work.

Note: the problem is worth 25 points if your result has $O(n \log n)$ complexity, and is worth 50 points if the complexity is $O(n)$. Please submit the solution to this problem separately.