Problem Set 9

All parts are due on November 29, 2018 at 11PM. Please write your solutions in the LaTeX and Python templates provided. Aim for concise solutions; convoluted and obtuse descriptions might receive low marks, even when they are correct. Solutions should be submitted on the course website, and any code should be submitted for automated checking on `alg.mit.edu`.

Problem 9-1. [28 points] **Tree Counting**

How many heaps, binary search trees, and AVL trees are there on n distinct integer keys? In this problem, you will describe an algorithm to compute each of these counts. Instead of analyzing each algorithm's running time, please report the asymptotic number of **arithmetic operations** performed by your algorithm (additions, subtractions, multiplications, and divisions).[1]

(a) [8 points] Recall, a **Binary Search Tree** is a binary tree with a key at each node, satisfying the BST property. There are 5 binary search trees on the $n = 3$ distinct keys $\{60, 0, 6\}$. Describe a dynamic-programming algorithm to compute the number of different binary search trees on a given set of n distinct integer keys, using at most $O(n^2)$ arithmetic operations.

(b) [10 points] Recall, an **AVL Tree** is a binary tree with a key stored at each node, satisfying both the BST and AVL properties. There is only 1 AVL tree on the $n = 3$ distinct keys $\{60, 0, 6\}$. Describe an dynamic-progamming algorithm to compute the number of different AVL trees on a given set of n distinct integer keys, using at most $O(n^2 \log n)$ arithmetic operations.

(c) [10 points] Recall, a **Binary Min Heap** is a left-justified, complete binary tree with a key at each node, satisfying the min-heap property. There are 2 binary min heaps on the $n = 3$ distinct keys $\{60, 0, 6\}$. Describe a dynamic-programming algorithm to compute the number of different binary min heaps on a given set of n distinct integer keys, using at most $O(n)$ arithmetic operations.

[1] Numbers in this problem can grow too large to fit in a constant number of machine words, so arithmetic can take more than constant time. Thus instead, you should report the asymptotic number of arithmetic operations performed.

Problem 9-2. Treasureship! [10 points]

The new boardgame Treasureship is played by placing 2×1 ships within a $2 \times n$ rectangular grid. Just as in regular battleship, each 2×1 ship can be placed either horizontally or vertically, occupying exactly 2 grid squares, and each grid square may only be occupied by a single ship. Each grid square has a positive or negative integer value, representing how much treasure may be acquired or lost at that square. You may place as many ships on the board as you like, with the score of a placement of ships being the value sum of all grid squares covered by ships. Design an efficient dynamic-programming algorithm to determine a placement of ships that will maximize your total score.

Problem 9-3. Circleworld Politics [15 points]

Circleworld, an essentially one-dimensional circular nation surrounding a sun, is home to n residents (where n is odd). There are two political parties in Circleworld: the Maryonettes and the Itnizks, with the Maryonettes currently in power. Each resident is a member of one of these two parties and always votes for their party in any election. Mary Jander, the leader of the Maryonettes, lives at address 1 in Circleworld. Each other resident lives at a unique consecutively increasing integer address starting at Mary's house going around the circular world in one direction, so that Mary's next door neighbor in that direction lives at address 2, and her other next door neighbor in the opposite direction lives at address n.

Circleworld is currently divided into d districts that are each **odd-contiguous**. A district is odd-contiguous if it contains an odd number of residents who live at a contiguous sequence of addresses along the circular world. For example, Mary currently lives in district 1, comprising 7 residents who live at addresses $(n-1, n, 1, 2, 3, 4, 5)$. When an election is held, the residents in each district cast a vote for their political party, and the party receiving the majority of votes within a district wins that district. Being the leader of the party in power, Mary is allowed to completely reassign the d districts in Circleworld prior to the upcoming election, under the condition that each of the new d districts is odd-contiguous, and that every resident is assigned to exactly one district. Given the address and political party of each resident in Circleworld, describe an efficient dynamic-programming algorithm to help Mary reassign districts so as to maximize the number of districts her party can win in the upcoming election.

Problem 9-4. [12 points] **Bottle Breaker Remix**

After witnessing the popularity of *Green Zombie Atonement II* (GZA2), a competitor has released a knockoff videogame called *Blue Demon Confessions III* (BDC3). BDC3 features the exact same bottle-breaking minigame as in GZA2 (from PS8) with one key difference: whenever one or more bottles are hit and shatter on the ground, the remaining bottles **get pushed together**, so that the bottles on either side of the bottles that just broke **become adjacent**. As in GZA2, you may throw a ball to hit any single bottle or any pair of adjacent bottles; but in BDC3, you may hit adjacent bottle pairs even if they were not adjacent in the original lineup, to score the product of their labels. For example, if the lineup of bottle labels is $(5, -3, -5, 1, 2, 9, -4)$, then one sub-optimal[2] score attainable is 38: you could throw balls at pairs $(-3, -5)$, $(2, 9)$, and $(5, 1)$ (which become adjacent after $(-3, -5)$ are hit); or, you could throw a ball at pair $(2, 9)$, then throw a ball at 1 to get rid of that bottle, and then throw a ball at pair $(-5, -4)$, to achieve the same score. Given a line of n bottle labels, describe an efficient dynamic-programming algorithm to determine the maximum possible score.

Problem 9-5. Hidden Palindromes [35 points]

Cayson wants to send hidden messages to his friends Ack and Dirk, via Derrit, an online public message board. Cayson will post to the message board a seemingly random string of lowercase letters, containing a secret message within a palindrome subsequence. A **palindrome** is any string which is the same string when the order of its characters is reversed. String B is a **subsequence** of a string A if the letters of B appear in order in A, possibly with other letters interspersed. Given an online message board post A, Cayson's secret message B will be the **first half** of the longest palindrome subsequence of A. If the longest palindrome subsequence has an odd number of characters, the secret message will include the middle character. You may assume that the longest palindrome subsequence of one of Cayson's online message board posts is unique. For example, if Cayson's message board post is `"nweeyaoslsoitmarawtuitfsjaipdiwi"`, the (unique) longest subsequence palindrome is `"wasitaratisaw"`[3], with the secret message being `"wasitar"`[4].

- **(a)** [10 points] Given a message board post from Cayson containing n characters, describe an $O(n^2)$-time algorithm to decode it and return Cayson's secret message.
- **(b)** [25 points] Implement the decoding algorithm described above in a Python function `decode_message`. You can download a code template containing some test cases from the website. Submit your code online at `alg.mit.edu`.

---

[2]The optimal score for this lineup is 62, achievable by hitting pairs $(-3, -5)$, $(1, 2)$, and $(5, 9)$.

[3]Was it a rat I saw?

[4]Was it AR?