# CS 124 Homework 4: Spring 2021

**Your name:**

**Collaborators:**

**No. of late days used on previous psets:**
**No. of late days used after including this pset:**

Homework is due Wednesday 2021-03-10 at 11:59pm ET. You are allowed up to **twelve** (college)/**forty** (extension school) late days through the semester, but the number of late days you take on each assignment must be a nonnegative integer at most **two** (college)/**four** (extension school).

Try to make your answers as clear and concise as possible; style will count in your grades. Be sure to read and know the collaboration policy in the course syllabus. Assignments must be submitted in pdf format on Gradescope. If you do assignments by hand, you will need to scan in your results to turn them in.

For all homework problems where you are asked to design or give an algorithm, you must prove the correctness of your algorithm and prove the best upper bound that you can give for the running time. Generally better running times will get better credit; generally exponential time algorithms (unless specifically asked for) will receive no or little credit. You should always write a clear informal description of your algorithm in English. You may also write pseudocode if you feel your informal explanation requires more precision and detail, but keep in mind pseudocode does NOT substitute for an explanation. Answers that consist solely of pseudocode will receive little or not credit. Again, try to make your answers clear and concise.

1. (a) **(20 points)** A challenge that arises in databases is how to summarize data in easy-to-display formats, such as a histogram. A problem in this context is the minimal imbalance problem. Suppose we have an array $A$ containing $n$ numbers, all positive, and another input $k$. Consider $k$ indices $j_1, j_2, \ldots j_k$ that partition the array into $k+1$ subarrays $A[1, j_1], A[j_1+1, j_2], \ldots, A[j_k+1, n]$. The weight $w(i)$ of the $i$th subarray is the sum of its entries. The *imbalance* of the partition is

$$\max_i \left| w(i) - \left( \sum_{\ell=1}^{n} A[\ell] \right) / (k+1) \right|.$$

That is, the imbalance is the maximum deviation any partition has from the average size.

Give an algorithm for determining the partition with the minimal imbalance given $A$, $n$, and $k$. (This corresponds to finding a histogram with $k$ breaking points, giving $k+1$ bars, as close to equal as possible, in some sense.)

(b) **(5 points)** Explain how your algorithm would change if the imbalance was redefined to be

$$\sum_i \left| w(i) - \left( \sum_{\ell=1}^{n} A[\ell] \right) / (k+1) \right|.$$

2. (a) **(20 points)** Suppose we want to print a paragraph neatly on a page. The paragraph consists of words of length $\ell_1, \ell_2, \ldots, \ell_n$. The maximum line length is $M$. (Assume $\ell_i \leq M$ always.) We define a measure of neatness as follows. The extra space on a line (using one space between words) containing words $\ell_i$ through $\ell_j$ is $M - j + i - \sum_{k=i}^{j} \ell_k$. The penalty is the sum over all lines **except the last** of the **cube** of the extra space at the end of the line. This has been proven to be an effective heuristic for neatness in practice. Find a dynamic programming algorithm to determine the neatest way to print a paragraph. Of course you should provide a recursive definition of the value of the optimal solution that motivates your algorithm.

(b) **(20 points)** Determine the minimal penalty, and corresponding optimal division of words into lines, for the following review of the Season 1 Buffy DVD, apparently written by Ryan Crackell for the Apollo Guide, for the cases where $M = 40$ and $M = 72$.

(You can find the answer by whatever method you like, but we recommend coding your algorithm from the previous part. You don't need to submit code.)

(The text of the review may be easier to copy-paste from the tex source than from the pdf. If you copy-paste from the pdf, check that all the characters show up correctly—the "ff" in "Buffy", among others, often doesn't.)

Buffy the Vampire Slayer fans are sure to get their fix with the DVD release of the show's first season. The three-disc collection includes all 12 episodes as well as many extras. There is a collection of interviews by the show's creator Joss Whedon in which he explains his inspiration for the show as well as comments on the various cast members. Much of the same material is covered in more depth with Whedon's commentary track for the show's first two episodes that make up the Buffy the Vampire Slayer pilot. The most interesting points of Whedon's commentary come from his explanation of the learning curve he encountered shifting from blockbuster films like Toy Story to a much lower-budget television series. The first disc also includes a short interview with David Boreanaz who plays the role of Angel. Other features include the script for the pilot episodes, a trailer, a large photo gallery of publicity shots and in-depth biographies of Whedon and several of the show's stars, including Sarah Michelle Gellar, Alyson Hannigan and Nicholas Brendon.

3. **(25 points)** Consider the following string compression problem. We are given a dictionary of $m$ strings, each of length at most $k$. We want to encode a data string of length $n$ using as few dictionary strings as possible. For example, if the data string is bababbaababa and the dictionary is (a,ba,abab,b), the best encoding is (b,abab,ba,abab,a). Give an $O(nmk)$ algorithm to find the length of the best encoding or return an appropriate answer if no encoding exists.

4. **(0 points, optional)**[1] Consider an algorithm for integer multiplication of two $n$-digit numbers where each number is split into three parts, each with $n/3$ digits.

   (a) Design and explain such an algorithm, similar to the integer multiplication algorithm presented in class. Your algorithm should describe how to multiply the two integers using only six multiplications on the smaller parts (instead of the straightforward nine).

   (b) Determine the asymptotic running time of your algorithm. Would you rather split it into two parts or three parts?

   (c) Suppose you could use only five multiplications instead of six. Then determine the asymptotic running time of such an algorithm. In this case, would you rather split it into two parts or three parts?

   (d) (Harder; the instructors will be impressed if you solve it.) Find a way to use only five multiplications on the smaller parts. Can you generalize to when the two initial $n$-digit numbers are split into $k$ parts, each with $n/k$ digits? Hint: also consider multiplication by a constant, such as 2; note that multiplying by 2 does not count as one of the five multiplications. You may need to use some linear algebra.

[1] We won't use this question for grades. Try it if you're interested. It may be used for recommendations/TF hiring.