CS 315 Week 3 (Feb 12 and 14) summary and review questions

Topics covered

- Analysis of algorithms
- List ADT (Abstract Data Type), some list functions (reverse), concept of static function, algorithm for generating all the subsets of a given set of integers.

SUMMARY:

- Analysis of algorithms is often done before implementation – to compare different approaches and select the best one to implement.
- Measures of performance: code size, simplicity, memory requirement, number of instructions executed. We will use the last one as the primary measure of performance.
- Best, worst and average cases. Often, worst-case is used to measure the performance of an algorithm.
- Order notation is a good approximation to the estimate, and we keep only the most significant term in the expression.
- Maximum subsequence sum problem – we considered three algorithms taking time $O(n^3)$, $O(n^2)$ and $O(n)$. On an input of size one million, the first algorithm may take several years while the last one would take less than 1 second on a standard desktop computer.
- Discussion of image representation using BMP format:
    - Color components – R, G and B with 8 bits each
    - Uncompressed format
- Algorithm to remove salt and pepper noise – mean filter. Time complexity is O(N) where N = the number of pixels in the image. But the constant involves is rather high, about 30. This can be improved slightly.
- Median-filter : Another filter that avoids the blurring caused by mean-filter.
- Abstract data types – its advantages and disadvantages.
- ADT list
- Some list functions – we examined how to reverse a list without creating new nodes.

- Concept of static function
- Code for build for a problem similar to the combinations problem of project 1. (subset generation.)

Review questions:

1) Chapter 2 contains several exercises on algorithm analysis. Work on them – specifically Exercise 2.7 and other similar ones.

2) Download EasyBMP package (you can find it at: http://easybmp.sourceforge.net/download.html Then, you can compile a program (e.g. myProgram.cpp) as follows:
   % g++ myProgram.cpp EasyBMP.cpp -o myOutput
   (assuming EasyBMP.cpp is in the same directory as myProgram.cpp).
   Create an image file in bmp format. (You can do this using any image viewing program or paint as follows: open the image in the viewer and add some random salt and pepper nose, then save choosing the bmp format.) Copy the program mean_filter.cpp from the web page http://ravi.cs.sonoma.edu/cs31sp08/Lab/index.html, and test it with the image you created.

3) Implement the median-filter we discussed in class using selection sorting as well as insertion sorting to find the median. Compare the two programs in terms of the time taken to filter the image. Test the same image (in (2) above) and compare the two outputs.

4) Given an array of 0's and 1's, we want to find the longest sequence of consecutive 1's in it. For example, for the input array < 0, 1, 1, 1, 0, **1, 1, 1, 1**, 0, 1, 1, 0, 1, 1, 0 >, the maximum such sequence is of length 4. Describe an algorithm for this problem and obtain the number of operations performed in terms of n, the size of the array. Express your answer using O notation.

5) * Consider the two-dimensional generalization of the previous problem: You are given a black and white n x n image. (Every pixel is either white or black.) You are to find the largest black square region in this image. Describe an algorithm to solve this problem and estimate its time complexity using O notation.

6) Write the reverse function (for a standard singly-linked list) using only the following operations: **`size(), first(), remove(int) and insertFirst(object).`**
   The function **`remove(k)`** removes the object in position k of the list. What is the complexity of your algorithm?