CS 140: Algorithms(Fall 2016)

Homework 4

Due time: 2016-11-23

Part1 Handwriting:

1. (20%) We have talked about the 0-1 Knapsack Problem in recitation. For this problems, you are given n objects and a knapsack, Item i weighs $w_i > 0$ kilograms and has value $v_i > 0$. Knapsack has capacity of W kilograms. The goal: fill knapsack so as to maximize total value.
   (a) Write the state transition equation.
   (b) If we have the input as following and W=11, fill the table.

Item	Value	Weight
1	1	1
2	6	2
3	18	5
4	22	6
5	28	7

	0	1	2	3	4	5	6	7	8	9	10	11
Ø												
{1}												
{1,2}												
{1,2,3}												
{1,2,3,4}												
{1,2,3,4,5}												

   (c) Write down the time complexity and space complexity. Could you optimize them? Please write your method down.

2. (15%) You are given a string of characters $S = s_1,s_2,...,s_n$ where all non-alphabetic characters have been removed (e.g. “thisisasentencewithoutanyspacesorpunctuation”) and a function dict(w,i,j), which takes as input a string w and two indices i and j and returns true if the string wi...j is a dictionary word and false otherwise.
   (a) Give a solution that determines whether the string S consists of a sequence of valid dictionary words. Analyze the time complexity. Assume calls to dict are O(1).
   (b) If the string is valid, describe how to modify your algorithm to output the sequence of words. Analyze the time complexity.

3. (15%) You are given an arithmetic expression containing n real numbers and n−1 operators, each either + or ×. Your goal is to perform the operations in an order that maximizes the value of the expression. That is, insert parentheses into the expression so that its value is maximized.
   For example:
   • For the expression 6×3 + 2×5, the optimal ordering is to add the middle numbers first, then perform the multiplications: (6 × (3 + 2)) × 5 = 150.
   • For the expression 0.1 × 0.1 + 0.1, the optimal ordering is to perform the multiplication first, then the addition:

(0.1 × 0.1) + 0.1 = 0.11.

- For the expression (−3) × 3 + 3, the optimal ordering is ((−3) × 3) + 3) = −6.

(a) Clearly state the set of subproblems that you will use to solve this problem.

(b) Write a recurrence relating the solution of a general subproblem to solutions of smaller subproblems.

(c) Analyze the running time of your resulting dynamic programming algorithm, including the number of subproblems and the time spent per subproblem.

Part2 Programming: (50%)

Please hand in C/C++ code and images to **algorithms_2016@163.com** with subject:
CS140HW4_StudentID_Name (WRONG subject will be rejected and not to be graded)
before 24:00, November 23th.

In this part, your program's input and output are images. We have provided Matlab code (IMG2TXT.m, TXT2IMG.m) to convert image and txt file for your convenience (Or you can use come third party libraries to do these work). Please use them properly.

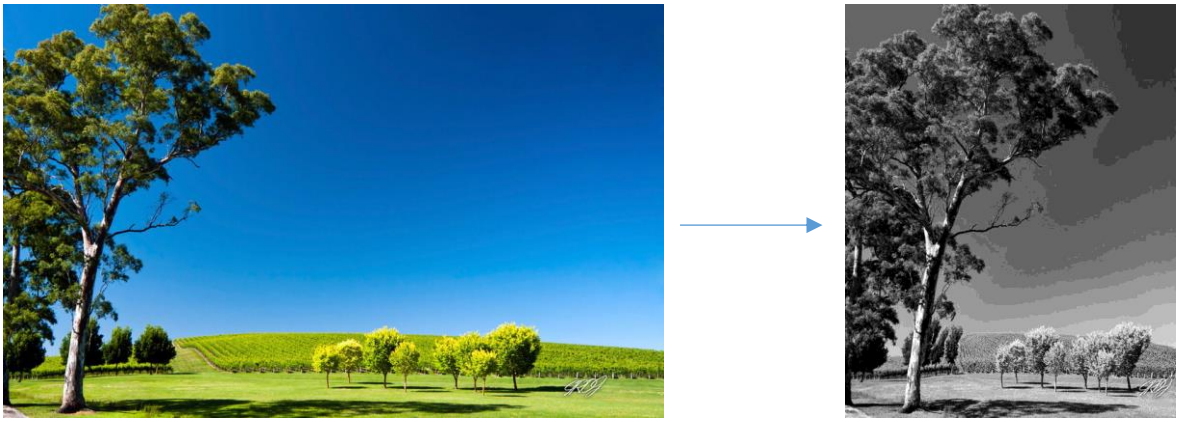
For undergraduate:

Seam Carving.

Please view this link: https://en.wikipedia.org/wiki/Seam_carving

Using Dynamic Programming to solve this problem.

Your output image's width should be reduced by half.



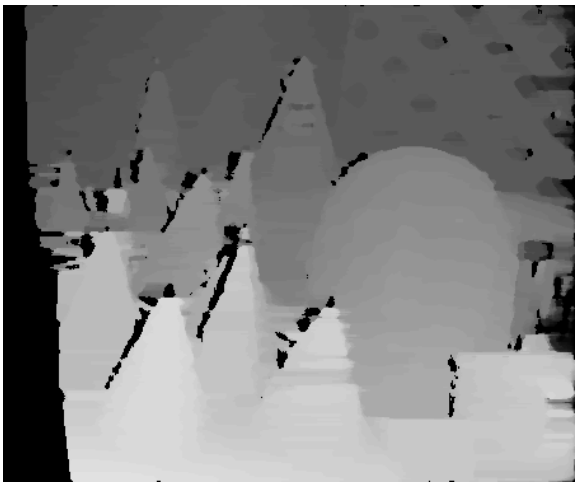
There are many methods to compute the energy. You can try them and get a better result.

Please process "data2.jpg" and submit the output "out.jpg" and your code. Try to optimize the result, we will give you extra credits. :)

For graduate:

Stereo matching.

For this problem set you will solve the stereo correspondence problem using dynamic programming, as described in class. The goal of this algorithm is to find the lowest cost matching between the left and right images.



Please process “L.png” and “R.png” and submit the output “out.jpg” and your code. Try to optimize the result, we will give you extra credits. :)