CSE421: Design and Analysis of Algorithms	Feb 22, 2018
Homework 7	
Shayan Oveis Gharan	Due: March 1st, 2018 at 5:00 PM

Please see https://courses.cs.washington.edu/courses/cse421/18wi/grading.html for general guidelines about Homework problems.

Most of the problems only require one or two key ideas for their solution. It will help you a lot to spell out these main ideas so that you can get most of the credit for a problem even if you err on the finer details. Please justify all answers.

P1) When studying greedy algorithms we learned that the Greedy algorithm fails to return the minimum number of coins for a change. In this problem we want to use dynamic programming to solve that question. Suppose we have n types of coins, type i has weight $w[i]$. Furthermore, suppose we have an infinite number of coins of each time. We want to return a change of W. Design an algorithm that runs in time polynomial in n, W and returns the minimum number of coins needed to change W with the above coins.

For example if we have three types of coins with weight $1, 5, 7$ we can change 13 with 3 coins: $7 + 5 + 1$ and 10 with two coins $5 + 5$.

You can assume all entries of $w[i]$ are integers. Your algorithm can only use $O(W)$ many words of memory (Here we are assuming that each integer in the range $1, \ldots, W$ fits in a word of memory.

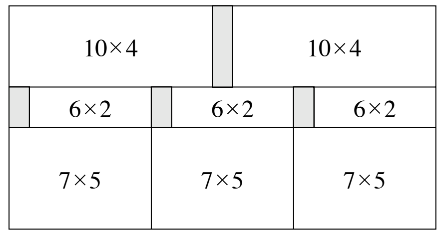
P2) A balanced-bracket string BBS is defined as follows:

- The empty string is a BBS
- If X, Y are BBSs then (X) and XY are BBSs
- Nothing else is a BBS

For example, $(), (()), ()(()())(())$ are BBSs while $((),)(, (()()))()($ are not. Use dynamic programming to design a polynomial time algorithm that for a given n, outputs the number of different BBSs of length n. For example for $n = 4$ you should output 2 corresponding to $()()$ and $(())$, and for $n = 5$ you should output 0.

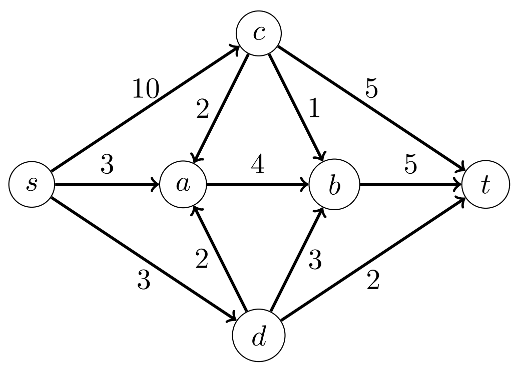
P3) Suppose we have received a large rectangular marble slab of size $W \times H$ and we want to cut the slab to obtain rectangular marble plates of sizes $W_1 \times H_1, W_2 \times H_2, \ldots, W_n \times H_n$. Any piece of marble (the slab or the plates cut from it) can be cut either horizontally or vertically into two rectangular plates with integral widths and heights, cutting completely through that piece. This is the only way to cut pieces and pieces cannot be joined together.

Since the marble has a pattern on it, the plates cannot be rotated: if a plate of size $A \times B$ is cut, then it cannot be used as a plate of size $B \times A$ unless $A = B$. We can make zero or more plates of each desired size. A marble plate is wasted if it is not of any of the desired sizes after all cuts are completed. The question is how to cut the initial slab so that as little of it as possible will be wasted.



As an example, assume that original slab is 21×11 and the desired plate sizes are $10 \times 4, 6 \times 2, 7 \times 5$, and 15×10. The minimum possible area wasted is 10, and the figure (below) shows one sequence of cuts with total waste area of size 10. Design an algorithm that runs in time polynomial in n, W, H that outputs the minimum possible waste area.

P4) Draw out a maximum s-t flow for the graph below, and the corresponding residual graph G_f. What is the minimum cut that corresponds to this max flow?



P5) **Extra Credit:** You are given an $m \times n$ array of real numbers. Suppose that the numbers in each row add up to an integer and the numbers in each column add up to an integer. You want to substitute each number $A[i,j]$ with $\lfloor A[i,j] \rfloor$ or $\lceil A[i,j] \rceil$ such that the sum of the numbers in each row and each column remain invariant. Design a polynomial time algorithm that outputs the integer array.

For example, if the input is the left table you can output the right table. Note the sum of numbers in each row (and each column) of the left table is the same as the sum of the numbers of the same row (resp. the same column) in the right table.

0.4	0.1	1.5
0.6	1.9	0.5

$\Rightarrow$

1	0	1
0	2	1