CSE421: Design and Analysis of Algorithms	May 12, 2022
Homework 6	
Shayan Oveis Gharan	Due: May 19, 2022 at 23:59 PM

P1) Consider the following instance of the knapsack problem: We are given n items with (integer) weights $w_1, \ldots, w_n$ and values $v_1, \ldots, v_n$ and knapsack of weight W. Furthermore, assume $w_i \leq W/2$ for all i. Design an algorithm that runs in time polynomial in n and $\log W$ and outputs a 2-approximation for the knapsack problem. In other words, your algorithm should return a set of items of total weight at most W and value at least half of the total value of OPT. Note that any basic operation (such as addition, multiplication, division, etc) on integers in the range $[0, W]$ takes poly$(\log W)$ times.

Hint: To upper-bound the value of OPT think of the algorithm for a modified version of the problem: Suppose that each item $1, \ldots, n$ is a divisible quantity, such as milk, sugar, etc. The algorithm keeps taking as much as possible from item that has the highest value (per ounce) until it gets the full weight, then goes to the second most valuable item, etc. It keeps doing this until the knapsack is full. It turns out that this greedy algorithm returns the optimum solution when the input is divisible. In our case that it in-divisible, you can use without proof that the value of this algorithm is an upper-bound to the value of OPT.

P2) You are given an $n \times n$ array A where for all $1 \leq i, j \leq n$, $A[i, j]$ is an integer that may be negative. For a rectangle $(x_1, y_1), (x_2, y_2)$ where $x_1 \leq x_2$ and $y_1 \leq y_2$, the value is the sum of all numbers in this rectangle, i.e.,

$$\sum_{i=x_1}^{x_2} \sum_{j=y_1}^{y_2} A[i, j]$$

Design an algorithm that runs in time $O(n^3)$ and outputs the value of the rectangle of largest value. Note that the value of the empty rectangle is zero. For example, if A is the following array, the optimum rectangle has value 8.

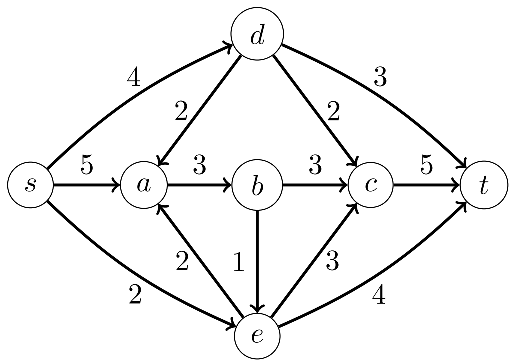
0	4	3
3	1	-5
-2	1	3

P3) Given a tree T with n vertices and an integer $k \geq 1$ such that every vertex of T has degree $deg(v) \leq 3$, we want to choose a set S of k vertices of the tree which are connected, i.e., for any pair of vertices $u, v \in S$ the unique path between u, v in T is also in S. Design a polynomial time algorithm that outputs the number of such sets S. For example given the following tree and $k = 3$ you should output 6 corresponding to the combinations $\{1, 2, 3\}, \{1, 2, 5\}, \{2, 3, 5\}, \{2, 4, 5\}, \{2, 5, 6\}, \{4, 5, 6\}$.

Also, for $k = 4$ should you output 6 corresponding to the combinations $\{1, 2, 3, 5\}$, $\{4, 5, 6, 2\}$, $\{1, 2, 4, 5\}$, $\{1, 2, 5, 6\}$, $\{2, 3, 4, 5\}$, $\{2, 3, 5, 6\}$.

Extra credit: Solve this problem for all possible trees.

P4) Draw out a maximum s-t flow for the graph below, and the corresponding residual graph G_f. You don't need to show your work. What is the minimum cut that corresponds to this max flow? Write down the capacity of the cut.



P5) **Extra Credit:** Given a sequence of positive numbers $x_1, \ldots, x_n$ and an integer k, design a polynomial time algorithm that outputs

$$\sum_{S \in \binom{n}{k}} \prod_{i \in S} x_i,$$

where the sum is over all subsets of size k.