Name: ______________________________

CSE 5500 Algorithms, Spring 2018

Exam II-B Solutions

1. First compute $f_i(x) = (x + a_i)^{2^i}$, for $1 \leq i \leq \log n$. For any i, $(x + a_i)^{2^i}$ can be computed in time $O(2^i)$ (as per Problem 1, Homework 2). Thus all of these polynomials can be computed in a total of $O\left(\sum_{i=1}^{\log n} 2^i\right) = O(n)$ time.

   Followed by this, we do the following:

   $f(x) = f_1(x)$;
   for $i = 2$ **to** $\log n$ **do**
   $\quad f(x) = f(x) \times f_i(x)$;

   Using the theorem that we can multiply two degree d polynomials in $O(d \log d)$ time, the total run time will be $O(\sum_{i=1}^{\log n} 2^i i) = O(n \log n)$.

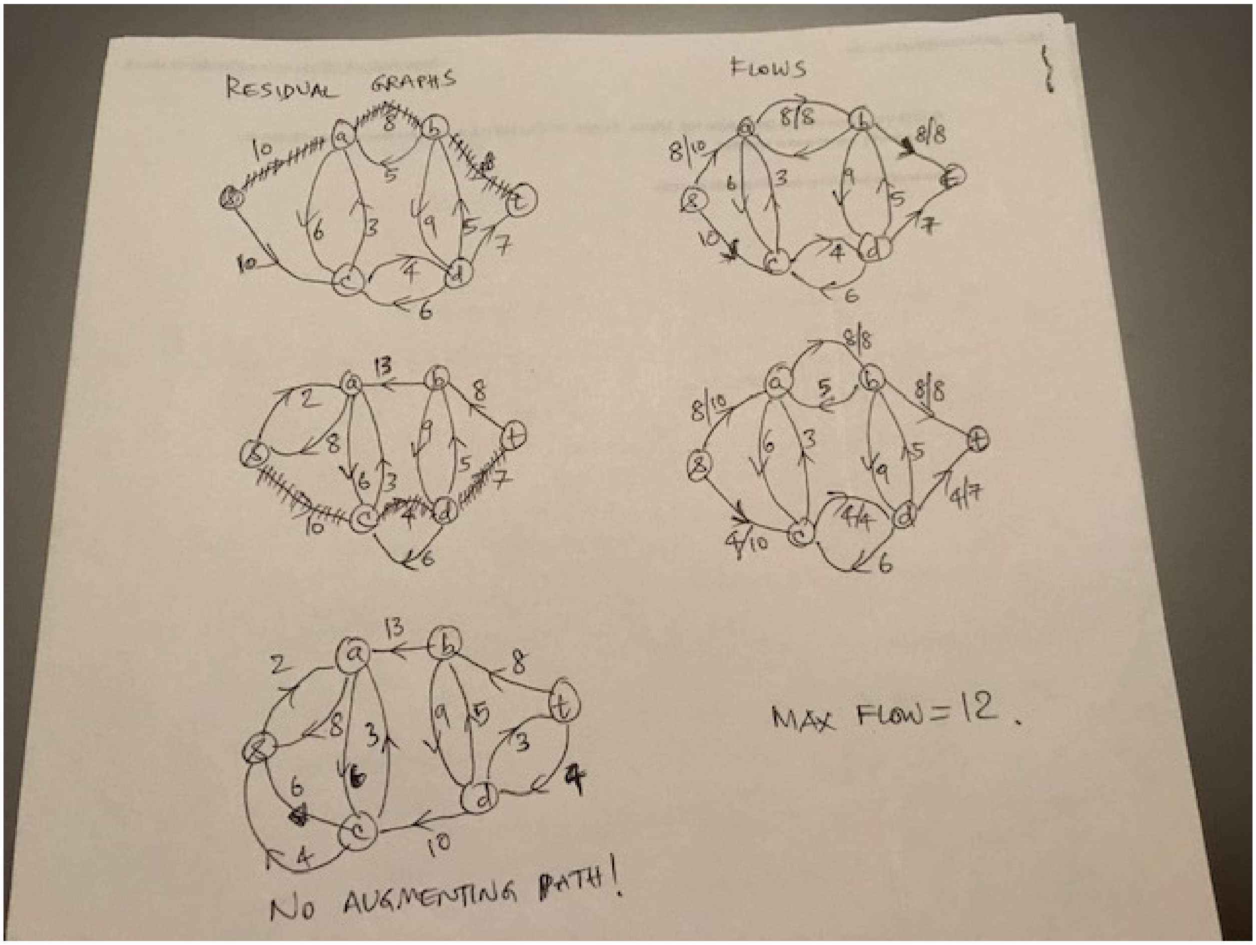
2. The occurrences of Y in X can be found by sliding Y under X and by checking if there is a complete match between Y and X in each of those positions. We need to check the match at $n + m + 1$ positions ($m + 1$ out of these $n + m + 1$ positions give only partial matches). To find a match between X and Y, both the 0's and 1's in X and Y have to match.

   Each of the bits in a binary string of length n can be treated as the coefficients of a polynomial of degree n. The multiplication of the polynomials of X and Y created as above (from the binary strings) will contain $n + m + 1$ coefficients. Note that each one of those coefficients represent the number of matches of 1's between the strings X and Y at each of the $n+m+1$ positions (sliding Y from right to left). To find the number of matches of 0's between the strings, do the following: Change the 0's to 1's and 1's to 0's in both the strings and then do the multiplication again. Now, the sum of these two polynomials give the number of matches of 0's and 1's at each of the $n + m + 1$ positions. If any of the coefficient is m, that means there is a complete match at that position. And the number of matches can be found by doing a scan on the coefficients of the resulting polynomial.
   Complexity = Complexity of multiplying polynomials of degree n and $m = O(n \log n)$.

3. Any spanning tree will contain $(|V| - 1)$ edges. Since all the edges are of equal weight, the spanning tree returned by BFS (or DFS) will also be a MCST.
   Complexity = Complexity of BFS = $O(|V| + |E|) = O(|E|)$. Note here that the given graph is connected and hence $|E| \geq |V| - 1$.
   Total weight of the MCST = $w * (|V| - 1)$.

4. Since the edge weights are the same, the path from a node to any other node will be the shortest if the number of edges along this path is the smallest. Thus we can use BFS here. We perform BFS from every node u to figure out the shortest paths from u to every other node. The total run time = $O(|V|(|V| + |E|)) = O(|V|^2 + |V|\ |E|)$.

5. Let $minCount(x)$ return the minimum number of coins for the amount x.
$$\begin{aligned} minCount(x) \;=\; min\{&minCount(x - a_1) + 1, \\ &minCount(x - a_2) + 1, \\ &\vdots \\ &minCount(x - a_n) + 1\} \end{aligned}$$
   By computing $minCount(x)$ for each possible value of x, $1 \leq x \leq C$ and storing them in a table, we can see that the value of $minCount$ for each value can be derived from the $minCount$ values of n other entries. Complexity = $O(Cn)$.



6. Using Ford-Fulkerson algorithm we see that the max flow is 12.