HW3

Seyed Armin Vakil Ghahani
PSU ID: 914017982
CSE-565 Fall 2018
Collaboration with: Sara Mahdizadeh Shahri, Soheil Khadirsharbiyani
Muhammad Talha Imran

September 12, 2018

Problem 1. Sum of Functions

Solution

- a) For each i, $f_i(n) \in O(n)$. Hence, there is an N_i and c_i that inequality $f_i(n) \leq c_i * n$ is true for each $n \geq N_i$. Thus, for each $n \geq max_{n=1}^{m} N_i$:

$$s_m(n) \leq \sum_{i=1}^{m} c_i * n \Rightarrow s_m(n) \leq n * C$$

  which $C = \sum_{i=1}^{m} c_i$. Therefore, it proves that $s_m(n) \in O(n)$.

- b) For each i, $f_i(n) \in O(n)$. Hence, there is an N_i and c_i that inequality $f_i(n) \leq c_i * n$ is true for each $n \geq N_i$. Thus, for each $n \geq max_{i=1}^{n} N_i$:

$$s_n(n) \leq \sum_{i=1}^{n} c_i * n \Rightarrow s_n(n) \leq n * c_k * \sum_{i=1}^{n} \frac{c_i}{c_k}$$

  which $c_k = max_{i=1}^{n} c_i$. Hence, for each $1 \leq i \leq n$, $c_i \leq c_k \Rightarrow \frac{c_i}{c_k} \leq 1$. So,

$$s_n(n) \leq n * c_k * \sum_{i=1}^{n} 1 = n * c_k * n = n^2 * c_k$$

  Therefore, $s_n(n) \in O(n^2)$.

Problem 2. Shortest Cycles

Solution

- a) We are going to prove that the vertexes $v_i, ..., v_n, v_1, ..., v_{i-1}$ are on depth 0 through $k-1$ consecutively. It will be proven by contradiction. Suppose that, the first vertex that breaks this rule is v_{j+1} that is after v_j, and v_j and v_{j+1} are in depth $a \geq b$, consecutively. The length of path from v_i to v_{j+1} is b through the BFS tree, however, the length of this path through the cycle is $a+1$ which is greater than b (not equal). Hence, if we replace the path in the BFS tree with the path in the cycle, the length of the shortest cycle will be reduced that is in contradiction with the shortest cycle assumption.

- b) If we run a BFS algorithm from a node in the graph, it will give us the shortest cycle that concludes this vertex. Whenever a back edge to the root is shown in the BFS algorithm, it represents a cycle which includes root. Since, in BFS, the vertexes are traversed by their distance to the root, the first back edge to the root will be the shortest cycle. Hence, if we run this algorithm for every vertex in the graph, the shortest cycle of each vertex will be calculated and we can take the shortest cycle of the graph in O(n*(m+n)) because the BFS algorithm is O(m+n) and we run it for n times.