CSI - 3105 Design & Analysis of Algorithms
Fall 2021

Assignment #2

Deadline: Tuesday, October 19, 2021 before 10:00 AM (to be delivered in BrightSpace)

Note: Make sure your handwriting is readable, otherwise your assignment will not be marked.

1. (10 marks) Consider the following recurrence.

$$T(n) = \begin{cases} 5 & \text{if } 1 \leq n \leq 3, \\ 2n + T\left(\frac{n}{3}\right) + T\left(\frac{n}{4}\right) + T\left(\frac{n}{5}\right) & \text{if } n > 3. \end{cases}$$

   (a) (2 marks) What is the solution to this recurrence? Your answer must be as precise as possible. You must write your answer using the big-O notation.

   (b) (8 marks) Prove that your answer is correct. You must use a proof by induction. You cannot use the Master Theorem. You must justify every step of your proof.

Solution:

(a) (2 marks) $T(n) = O(n)$.

(b) (8 marks) We prove, by induction on n, that there exists a constant c such that $T(n) \leq c\,n$ for all $n \geq 1$.

Base Case: When $n \leq 3$,

$$T(1) = 5 \leq c \cdot 1,$$
$$T(2) = 5 \leq c \cdot 2,$$
$$T(3) = 5 \leq c \cdot 3,$$

provided that $c \geq 5$.

Induction Hypothesis: Let $n > 3$ be an integer and assume that

$$T(m) \leq c\,m$$

for all $1 \leq m < n$.

Induction Step: We have

$$\begin{aligned}T(n) &= 2n + T\left(\frac{n}{3}\right) + T\left(\frac{n}{4}\right) + T\left(\frac{n}{5}\right)\\ &\leq 2n + c\cdot\frac{n}{3} + c\cdot\frac{n}{4} + c\cdot\frac{n}{5} && \text{by the induction hypothesis,}\\ &= 2n + c\cdot\frac{47}{60}n\\ &\leq c\,n,\end{aligned}$$

provided that

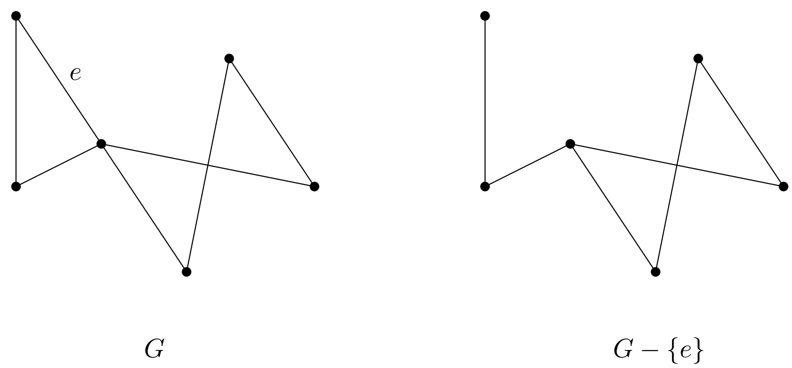
$$2 + c\cdot\frac{47}{60} \leq c,$$

i.e.

$$c \geq \frac{120}{13}.$$

By taking $c = \frac{120}{13}$, both the base case and the induction step work, which concludes the proof. □

2. (10 marks) Let $G = (V, E)$ be an undirected graph and $e \in E$ be an edge of G. We denote by $G - \{e\}$ the graph obtained from G by deleting e. Here is an example.



An edge $e = \{a, b\} \in E$ is said to be *critical* if there is no path between a and b in $G - \{e\}$.

Design a deterministic algorithm to solve the following problem.

input: An undirected graph $G = (V, E)$, stored using adjacency lists, together with an edge $e = \{a, b\} \in E$.

output: **CRITICAL** if e is critical. Otherwise, return **NOT**.

Your algorithm must be deterministic. Your algorithm must take $O(|V| + |E|)$ time. You must describe your algorithm in plain English (no pseudocode) and you must explain why the running time of your algorithm is $O(|V| + |E|)$.

Solution: Run the algorithm studied in class to identify the connected components in G. This takes $O(|V| + |E|)$ time.

Scan all the vertices to find the largest ccnumber in G. Let k be this number (this corresponds to the number of connected components in G). This takes $O(|V|)$ time.

Scan the adjacency list of a to remove b. Then scan the adjacency list of b to remove a. This takes $O(\deg(a) + \deg(b))$ time. We now have $G - \{e\}$.

Run the algorithm studied in class to identify the connected components in $G - \{e\}$. This takes $O(|V| + |E|)$ time.

Scan all the vertices to find the largest ccnumber in $G - \{e\}$. Let k' be this number (this corresponds to the number of connected components in $G - \{e\}$). This takes $O(|V|)$ time.

If $k = k'$, then return **NOT**. Otherwise, return **CRITICAL**. This takes $O(1)$ time.

In total, it takes

$$
\begin{aligned}
& O(|V| + |E|) + O(|V|) + O(\deg(a) + \deg(b)) + O(|V| + |E|) + O(|V|) + O(1) \\
= & O(|V| + |E|) + O(|V|) + O(2|V|) + O(|V| + |E|) + O(|V|) + O(1) \\
= & O(|V| + |E|)
\end{aligned}
$$

time.