CS 580 Algorithm Design, Analysis, And Implementation
Fall 2021 Vassilis Zikas HW 1

Due Wednesday Sept. 15 at 11:59 p.m.

1. **Asymptotic analysis.**

   (a) Rank the following functions (representing running times) from smallest to largest (in terms of growth with respect to n).

   $$n!, \log n, \ln n, 3^n, n^n, n^4, 2n^2+n, \log(4n^3)$$

   Group together functions in the same asymptotic class. You do not need to show any work/ explain your answers.

   (b) Suppose that $f(n)$ and $g(n)$ are positive function always greater than 1 for $n \geq 2^{200}$. Let $h(n) = f(n) + g(n)$. Prove or disprove: $h(n) \in \Theta(\max\{f(n), g(n)\})$.

   (c) The n^{th} Harmonic number is the sum of the reciprocal of the first n natural numbers i.e.

   $$H_n = \sum_{i=1}^{n} \frac{1}{i}$$

   Show that H_n is $\Theta(\log n)$.

Answer: Contributed by Himanshi.

(a) he functions in from smallest to largest is as follows:

   i. $\log n, \ln n, \log(4n^3)$
   ii. $2n^2 + n$
   iii. n^4
   iv. 3^n
   v. $n!$
   vi. n^n

(b) Let $N = 2^{200}$. Note that for any $n > N$ we have $h(n) = f(n) + g(n) \leq 2 \cdot \max\{f(n), g(n)\} = c \cdot \max\{f(n), g(n)\}$ where $c = 2$. Thus, by definition $h(n) \in O(\max\{(f(n), g(n)\})$. Also, $h(n) = f(n) + g(n) \geq \max\{f(n), g(n)\} = c' \cdot \max\{f(n), g(n)\}$ So, $h(n) \in \Omega(\max\{f(n), g(n)\})$. Put together, this means $h(n) \in \Theta(\max\{f(n), g(n)\})$ And so, this conjecture is **true**.

(c) *Approach 1.* We use the idea of bounding a series by integrals as done in Left/ Right Riemann sums (see `https://en.wikipedia.org/wiki/Riemann_sum`link).
We first show an upper bound on H_n by bounding the sum using a integral,

$$\begin{aligned}\sum_{i=2}^{n} \frac{1}{i} &\leq \int_{1}^{n} \frac{1}{x} dx \\ &= [\log x]_1^n \\ &= O(\log n)\end{aligned}$$

Thus, H_n is $O(\log n)$. Next we show the lower bound,

$$\begin{aligned}\sum_{i=1}^{n} \frac{1}{i} &\geq \int_1^n \frac{1}{x} dx \\ &= \Omega(\log n)\end{aligned}$$

Thus, H_n is $\Theta(\log n)$.
Approach 2. Consider the following lower bound on H_n,

$$\begin{aligned}H_n &= 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} \ldots + \frac{1}{n} \\ &\geq 1 + \frac{1}{2} + \left(\frac{1}{4} + \frac{1}{4}\right) + \left(\frac{1}{8} + \frac{1}{8} + \frac{1}{8} + \frac{1}{8}\right) + \ldots \\ &= 1 + \frac{1}{2} + \frac{1}{2} + \frac{1}{2} + \ldots + \frac{1}{2} && (\log n \text{ terms}) \\ &\geq \frac{1}{2} \log_2 n.\end{aligned}$$

Similarly consider the term-wise upper bound,

$$\begin{aligned}H_n &= 1 + \frac{1}{2} + \frac{1}{3} + \frac{1}{4} + \frac{1}{5} + \frac{1}{6} + \frac{1}{7} + \frac{1}{8} \ldots + \frac{1}{n} \\ &\leq 1 + 1 + \left(\frac{1}{2} + \frac{1}{2}\right) + \left(\frac{1}{4} + \frac{1}{4} + \frac{1}{4} + \frac{1}{4}\right) + \ldots && (\log n \text{ terms}) \\ &\leq 2 \log_2 n.\end{aligned}$$

Thus, $H_n \in \Theta(\log n)$.

2. **Graph Traversal.** Unlike countries nowadays which have complicated transportation systems, there is a country which is formed in a tree shape by railways. This means that in this country there are n cities and $n-1$ railways between cities. Now the president of this country would like to know the longest railway path in this country.

   (a) Design and analyze an algorithm to find the longest path in $O(n^2)$.

   (b) Prove the following statement: suppose that the longest railway path in this country starts from city a to city b, then for any given city x, the longest railway path starting from x is either from x to a or from x to b.

   (c) Design and analyze an algorithm to find the longest path in $O(n)$.

   Answer: Contributed by Jiacheng.

   (a) For every city x, use BFS to find the farthest city. Comparing all the longest paths you find for each city, you can find the longest path. Time complexity $O(n^2)$.

   (b) We use $dist(a,b)$ to represent the distance between city a and city b. $path(a,b)$ is to denote all the cities included in the path from city a to city b. We use contradiction here to prove the statement above.
   Assumption 1: there exists a city x which is able to find another city c so that $dist(x,c) > max(dist(x,a), dist(x,b))$
   Now we will divide the proof into 4 cases.
   Case 1: We assume that x and c is included in $path(a,b)$. In this case, $dist(x,c)$ should be smaller than $max(dist(x,a), dist(x,b))$ since c must be included in $path(x,a)$ or $path(x,b)$. However, we know $dist(x,c) > max(dist(x,a), dist(x,b))$, so we find a contradiction for this case.

Case 2: We assume that x is included in $path(a,b)$ but c is not included. In this case, we have $dist(x,c) + max(dist(x,a), dist(x,b)) > dist(x,a) + dist(x,b)$. So, the longest path is either from c to a or from c to b, but not from a to b. Another contradiction found.

Case 3: We assume that x is not included in $path(a,b)$ but c is included. In this case, c must be included in either $path(x,a)$ or $path(x,b)$, so $dist(x,c) < max(dist(x,a), dist(x,b))$, which is a contradiction to $dist(x,c) > max(dist(x,a), dist(x,b))$.

Case 4: We assume that both x and c are not included in $path(a,b)$. In this case, we assume that city z is included in $path(a,b)$ and $dist(x,z) <= min(dist(x,y)), \forall y \in path(a,b)$. City z is essentially the city in $path(a,b)$ that connects city x and city a(or b). There are two different cases for z: z is included in $path(x,c)$ or not included in $path(x,c)$. For the case where z is included in $path(x,c)$, then we can have $dist(x,c) - dist(x,z) = dist(z,c) > max(dist(z,a), dist(z,b))$, since the graph is a tree. Thus, the longest path is either from c to a or from c to b, but not from a to b. Another contradiction found. For the case where z is not included in $path(x,c)$, the longest path should go in this direction: $c \to x \to z \to a$ or $c \to x \to z \to b$, since $dist(c,z) = dist(c,x) + dist(x,z) > max(dist(z,a), dist(z,b))$. So, the longest path is either from c to a or from c to b, but not from a to b. Another contradiction found.

In conclusion, we are able to find a contradiction for every possible case of the assumption 1. This means this assumption does not hold. Thus, the statement described in sub-question(b) is correct.

(c) We just need to BFS twice. For the first BFS, we can start from any city in the graph and we need to find the farthest city a. Since we have proved the statement in (b), we can claim that this city a should be one end of the longest railway path. For the second BFS, since we know that the city a is one end of the longest path, we can simply start from city a and use BFS to find the farthest city b. The path from a to b is the longest path. The overall time complexity is O(n).

3. **Proof Techniques.**

(a) Lil Omega shared a conjecture with Lil Delta. Lil Delta claimed to prove the conjecture and emailed the proof to Lil Omega. Lil Omega is not good at proofs and is asking for help from the experts. The content of the email from Lil Delta is given below. Should Lil Omega accept the proof? Provide reasoning for your answer.

Theorem. *All students have the same eye color.*

Proof. We prove the statement by induction.
Since in the statement of the theorem there is no variable to induct on, we restate the theorem as follows.
$P(n)$: In any set of n students, all students have the same eye color.
Base case: $P(1)$ is true. In any set of one student, the student has the same eye color.
Inductive step: We assume $P(n)$ is true to prove $P(n+1)$.
Let the $n+1$ students be $S_1, S_2, ..., S_n, S_{n+1}$.
By our inductive hypothesis, any set of n students have the same eye color.
Therefore, the first n students $S_1, S_2, ..., S_n$ have the same eye color.
Similarly, the last n students $S_2, ..., S_n, S_{n+1}$ have the same eye color.
From preceding arguments, the eye color of S_1 is same as the eye color of $S_2, ..., S_n$, and the eye color of S_{n+1} is same as the eye color of $S_2, ..., S_n$.
It follows that eye colors of S_1, S_{n+1}, and $S_2, ..., S_n$ are same, i.e., $P(n+1)$ is true. □

(b) A full m-ary tree T ($m \geq 2$) is a rooted tree in which each node has either 0 or m children.

Let $d_T(x)$ denote the depth of a node x in T, which is the number of edges contained in the path from root of T to x.

Let $L(T)$ denote the sum of the depths of leaf nodes in T, i.e., $L(T) = \sum_{x \in Leaves(T)} d_T(x)$, and $I(T)$ denote the sum of the depths of internal nodes in T, i.e., $I(T) = \sum_{x \in NonLeaves(T)} d_T(x)$.

Using induction, prove that for a full m-ary tree T with n nodes, $L(T) = (m-1)I(T) + n - 1$.

Answer: Contributed by Ahammed.

(a) The inductive step is flawed. We have correctly established that $n > 1, P(n) \implies P(n+1)$. But, we have failed to establish $P(1) \implies P(2)$.

When $n = 1$, there is no student in the sequence of students $S_2, \ldots, S_n$. Therefore, the comparisons of S_1 to $S_2, \ldots, S_n$ and the comparison of S_{n+1} to $S_2, \ldots, S_n$ are meaningless.

(b) *Proof.* For a full m-ary tree T with n nodes, we want to prove following proposition.

$$P(n) : L(T) = (m-1)I(T) + n - 1$$

Base Case: $P(1)$ is true. For a tree T with 1 node, $L(T) = 0$ and $I(T) = 0$. Therefore, $L(T) = (m-1)I(T) + n - 1$.

Inductive Step: For $n > 1$ and $n' < n$, we assume $P(n')$ is true to prove $P(n)$.

Let T be a full m-ary tree with n nodes. Since $n > 1$, T must have an internal node x whose m children are all leaf nodes. Let T' be a tree obtained from T by replacing x and children of x with a leaf node. T' has $n - m$ nodes, and by the inductive hypothesis we get following.

$$L(T') = (m-1)I(T') + (n - m - 1) \tag{1}$$

We can relate $L(.)$ and $I(.)$ values of T and T' as follows.

$$L(T) = L(T') - d_T(x) + m(d_T(x) + 1) \tag{2}$$

$$I(T') = I(T) - d_T(x) \tag{3}$$

Substituting expression of $L(T')$ from Equation 1 in Equation 2 we obtain following.

$$L(T) = (m-1)I(T') + (n - m - 1) - d_T(x) + m(d_T(x) + 1)$$

Substituting expression of $I(T')$ from Equation 3 in preceding equation we get following.

$L(T) = (m-1)(I(T) - d_T(x)) + (n - m - 1) - d_T(x) + m(d_T(x) + 1) = (m-1)I(T) + n - 1$ □

4. **2 points** Have you assigned pages on Gradescope? **Answer:** Yes!