HW1

October 1, 2024

1 Homework 1

1.1 Naiara Alonso Montes

1.1.1 Problem 1

For each pair of functions, indicate which of the following relations hold: $f(n) = O(g(n))$, $f(n) = o(g(n))$, $f(n) = \Omega(g(n))$, or $f(n) = \Theta(g(n))$. Justify your answer.

(a) $f(n) = n^2 + n - 100$ and $g(n) = 100n^2 + 1000$;

(b) $f(n) = \frac{1}{n} + 5$ and $g(n) = 1$;

(c) $f(n) = n \log_2 n$ and $g(n) = \frac{n^2}{\sqrt{n} \log_2 n}$;

(d) $f(n) = 5^n$ and $g(n) = 2^{2n}$.

I will solve this relations using limits:

Case a

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n^2 + n - 1000}{100n^2 + 1000}$$

I will divide both numerator and denominator by n^2

$$\lim_{n\to\infty} \frac{1 + \frac{1}{n} - \frac{100}{n^2}}{100 + \frac{1000}{n^2}}$$

All fractions in the form of $\frac{a}{n^i}$ tend to 0. So:

$$\lim_{n\to\infty} \frac{1}{100} = \frac{1}{100}$$

Solution: $f(n) = \Theta g(n))$

Case b

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{\frac{1}{n} + 5}{1}$$

Fraction $\frac{1}{n}$ tends to 0. So:

$$\lim_{n\to\infty} \frac{5}{1} = 5$$

Solution: $f(n) = \Theta(g(n))$

Case c

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{n\log_2 n}{\sqrt{2}\log_2 n}$$

Multiply the complement of the denominator and simplify:

$$\lim_{n\to\infty} \frac{n\log_2 n \cdot \sqrt{2}\log_2 n}{n^2} = \lim_{n\to\infty} \frac{n^{\frac{3}{2}}(\log_2 n)^2}{n^2} = \lim_{n\to\infty} \frac{(\log_2 n)^2}{\sqrt{2}}$$

As denominator domines over numerator for all values of n:

$$\lim_{n\to\infty} \frac{(\log_2 n)^2}{\sqrt{2}} = 0$$

Solution: $f(n) = o(g(n))$

Case d

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = \lim_{n\to\infty} \frac{5^n}{2^{2n}}$$

We take common factor to a^n. So:

$$\lim_{n\to\infty} (\frac{5}{4})$$

By rule $\lim_{n\to\infty} a^n = \infty$

Solution: $f(n) = \Omega(g(n))$

1.2 Problem 2

Let $G(V, E)$ be an undirected connected finite graph. Consider weight function $w_1 : E \to (1, +\infty)$ defined on the edges of G. Let $T_1(V, E_1)$ be a spanning tree of G which has a minimmun weight with respect to the weight function $w_1(e)$.

Define a new weight function $w_2 : E \to \mathbb{R}^+$ such that $w_2(e) = 2w_1(e) - 1$ for every $e \in E$. Prove that $T_1(V, E_1)$ has a minimum weight (among all spanning trees) with respect to the weight function $w_2(e)$.

- Edges weight function w_2 is a linear transformation.
- A it is a linear increasing function, the weight order is maintained after $w_2(e)$ for all $a \in E$.
- The order of edges will be maintained and the minimun spanning tree will be the same.

Circuit free

- T_1 is a tree so it must be circuit free, by definition.

Connected

- T_1 is a tree so it must be connected, by definition.

Spanning Tree

- As T_1 it is a subgraph of G and contains all vertices V, by definition it must be a spanning tree.

Minimmun Spanning Tree

- Let T_2 the MST of G.
- Let G_2 be the graph G with function w_2 applied for all edges.
- Let T_1 be the MST of G_2.
- By Kruskal Algorithm, all edges are added from smallest to highest value to a MST.
- If we apply function w_2 to T_2:

$$\sum_{e \in T_2} 2w_1(e) - 1 = \sum_{e \in T_1} w(e)$$

with function w just returning the weight of an edge

- As the sum of edges weight for $T2$ is equal to T_1, T_1 is a MST of G.

1.3 Problem 3

Let $G(V, E)$ be an undirected connected finite graph with the weight function $w : E \to \mathbb{R}^+$. Let T be a minimun spanning tree of G. Prove that exists a run of Prim's Algorithm that finds T.

I will use induction for this problem:

Base case - Prim's Algorithm starts with a single vertex. - Since T is a spanning tree, it must contain this vertex.

Induction - Assume that at some step, the set of A vertices already included in the tree is a subset sepanned by T, and all edges are added by Prim's Algorithm. - The next edge to be added must go from vertex of A to V A. - By cut property (proved here), the smallest weight edge crossing this cut must be on T. - Since Prim's Algorithm selects the minimun edge weight of this cut, it will select an edge that is part of T. - By induction, at every step of Prim's Algorithm, it will select an edge that it is part of T. Therefore, the entire tree T will be constructed by Prim's Algorithm.

[ ]:
```
# to pdf
!jupyter nbconvert --to pdf HW1.ipynb
```