Computational Topology, Homework 1

Rishi Borkar

due: 3 February 2022

This homework assignment should be submitted as a single PDF file to D2L.

General homework expectations:

- Homework should be typeset using LaTex.
- Answers should be in complete sentences, and make sense without seeing the question.
- You will not plagiarize, nor will you share your written solutions with classmates. (But, discussing the questions is highly encouraged).
- List collaborators at the start of each question using the `collab` command.
- Put your answers where the `todo` command currently is (and remove the `todo`, but not the word `Answer`).
- If you are asked to come up with an algorithm, you are expected to give an algorithm that beats the brute force (and, if possible, of optimal time complexity). With your algorithm, please provide the following:
  - *What*: A prose explanation of the problem and the algorithm, including a description of the input/output.
  - *How*: Describe how the algorithm works, including giving psuedocode for it. Be sure to reference the pseudocode from within the prose explanation.
  - *How Fast*: Runtime, along with justification. (Or, in the extreme, a proof of termination).
  - *Why*: Justify why this algorithm works. At a minimum, I expect a statement of the loop invariant for each loop, or recursion invariant for each recursive function.

CSCI 535 Problem 1-1

Collaborators on this problem: *Braeden Sopp*

Tasks Please do the following:

1. Write this homework in LaTex. (You can use this document as a starting point!) Note: if you have not used LaTex before and this is an issue for you, please contact me.
2. Update your photo on D2L to be a recognizable headshot of you.
3. Sign up for the class slack group.
4. Fill out the course survey (found on the syllabus).

CSCI 535 Problem 1-2

Collaborators on this problem: *Braeden Sopp*

Please read (or skim through) Eugenia Cheng's Quick Guide for Writing Proofs and describe one thing that you already do well in your academic writing / proof writing, and one thing that you will work on improving throughout during this class. You can reference additional resources, but be sure to provide full references, either as footnotes or as a proper bibtex citation!

Answer:

During this class, I intend to work on improving the initial definitions and the assumptions that are to be made before you start the proof. I have also been told that I handwave occasionally during my proofs which I need to work on. This mainly stems from me not having an intial proper understanding of the definitions. However I also believe that once the inital groundwork and assumptions as setup, I do pretty well with the rest of the proof including making use of proof by induction and contradiction.

CSCI 535 Problem 1-3

Collaborators on this problem: *Braeden Sopp*

Follow the Inkscape tutorial found here: `http://tavmjong.free.fr/INKSCAPE/MANUAL/html/SoupCan.html`. Then, make an inkscape figure illustrating any concept that you would like. Include the (PDF) images of both the soup can and your own figure as floating figures in your final write-up for this problem. Don't forget to add a captions (and to reference the figures!

Answer:

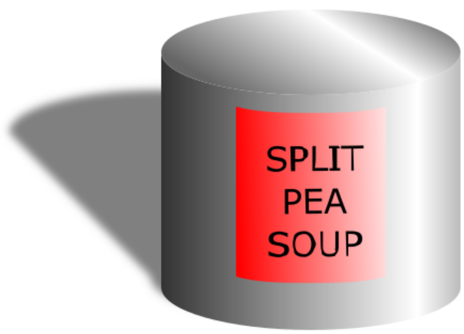

Figure 1: A Can of Soup
[1]

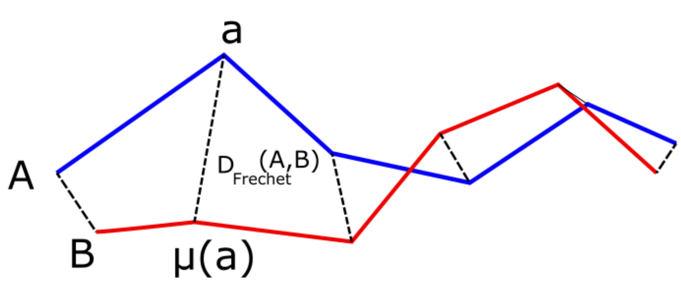

Figure 2: Frechet distance
[2]

CSCI 535 Problem 1-4

Collaborators on this problem: *Braeden Sopp*

Watch the short YouTube video. There is a topological error in the example he gives with the letter 'A'. What are the topological equivalence classes of the letters A seen in this video? Feel free to add additional items to your equivalence classes for the letter 'A'. (You don't need to prove that these are the equivalence classes, but a brief justification is expected).

Answer:

All of the "A's" in the video are topologically equivalent except the below two:

Figure 3: A_1

Figure 4: A_2

Two figures are topologically equivalent if one figure can be transformed into the other by twisting and stretching, but not tearing, cutting, or gluing. Since the above two A's would require cutting or gluing to be produced from the other A's, they belong to different topological classes compared to the other A's.

CSCI 535 Problem 1-5

Collaborators on this problem: *Braeden Sopp*

Use the definition of big-O notation to prove that $f(x) = n^2 + 3n + 2$ is $O(n^2)$.

Answer:

$f(n)$ is $O(g(n))$ if there are positive constants C and k such that:

$f(n) \leq Cg(n)$ whenever $n > k$

- Choose k = 1
- Assuming $n > 1$, then
- $\frac{f(n)}{g(n)} = \frac{n^2+3n+1}{n^2} < \frac{n^2+3n^2+n^2}{n^2} = \frac{5n^2}{n^2} = 5$
- Choose C = 5. Note that $3n < 3n^2$ and $1 < n2$.
- Thus, $n^2 + 3n + 1$ is On^2 because $n^2 + 3n + 1 \leq 5n^2$ whenever $n > 1$.

CSCI 535 Problem 1-6

Collaborators on this problem: *Braeden Sopp*

Let $f : \mathbb{R} \to \mathbb{R}$ be a function. The function f is considered continuous if for all open sets $A \subseteq \mathbb{R}$, $f^{-1}(A)$ is open in $\mathbb{R}$, where open is defined by the standard topology on $\mathbb{R}$ (so, what we normally think of as open). Let $c \in \mathbb{R}$. Consider the function $f_c : \mathbb{R} \to \mathbb{R}$ defined by $f_c(x) = c$ for all $x \in \mathbb{R}$. Prove that f_c is continuous.

Answer:

- Since $f_c : \mathbb{R} \to \mathbb{R}$ and $f_c(x) = c$, which means there exists a map $x \to c$.
- The image of $f[\{x\}] = c$
- The pre-image : $f^{-1}[\{c\}] = x$
- Since every sub-set of a discrete topological space is open, x is open. And therefore the pre-image of $f_c(x)$ is an open set as well.
- Hence, f_c is continuous.

CSCI 535 Problem 1-7

Collaborators on this problem: *Braeden Sopp*

Consider a graph $G = (V, E)$. The *eccentricity* of a vertex $v \in V$ is the maximum distance from v to any other vertex in V. (Note: the *(graph) distance* between $a, b \in V$ is the minimum length of a path from a to b; the *length* of a path in an unweighted graph is the number of edges in the path). The following algorithm computes the ecentricity of a vertex in a graph. For the while loop, provide the loop invariant and prove that it is the loop invariant. In this algorithm, Q is a (minimum) priority queue.

Algorithm 1 Eccentricity(G,v)

Input: a connected graph $G = (V, E)$ such that $|V| \geq 2$; a vertex $v \in V$
Output: the eccentricity of v

```
For each vertex, add an attribute dist and set it to ∞.
v.dist ← 0.
Create a priority queue Q of vertices, where each vertex w ∈ V has priority w.dist.
maxdist ← ∞
while Q is not empty do
  w ←Q.pop()                ▷ Since Q is a priority queue, w.dist ≤ x.dist for all x ∈ Q
  maxdist ← w.dist
  for each edge (w, x) ∈ E such that x ∈ Q do
    x.dist ← min{x.dist, w.dist + 1}
  end for
end while
return maxdist
```

Answer:

Loop invariant: At the start of each iteration, the priority queue consists only of unvisited vertices and maxdist holds the value of the eccentricity of the sub-graph made up of only the visited vertices.

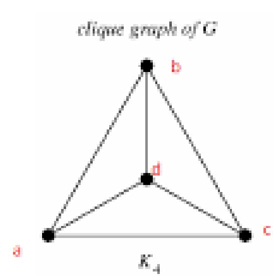

Figure 5: Graph

Initialization: At the start of the loop, the queue consists of all the vertices since none are visited and the maxdist = ∞ since the subgraph of visited vertices is 0

Maintenance:

maxdist = ∞	maxdist = 0	maxdist = 1	maxdist = 1	maxdist = 1
a = 0	~~a = 0~~	~~a = 0~~	~~a = 0~~	~~a = 0~~
b = ∞	b = 1	~~b = 1~~	~~b = 1~~	~~b = 1~~
c = ∞	c = 1	c = 1	~~c = 1~~	~~c = 1~~
d = ∞	d = 1	d = 1	d = 1	~~d = 1~~

Termination: The while-loop terminates when the queue is empty and the loop invariant gives the eccentricity of the entire graph. This is exactly the value that the algorithm should output, and which it then outputs. Therefore the algorithm is correct.

References

[1] Inkscape: Guide to a vector drawing program.

[2] Fréchet distance measurement.