CS 105: Introduction to Computer Science

Prof. Thao Nguyen

Spring 2025

Mid-semester Feedback Form

- Thank you to those who filled out the form!
- General workload/difficulty/course pace: 3 (medium)
- Things that are working/helpful:
  - Sample problems/exercises
  - Office/TA hours
- Room for improvement:
  - Confusing topics: loops, tuples, recursive design
  - More demos/practice work
  - Not enough preparations for labs

Logistics: Midterm 1

Class average: 60.9, standard deviation: 15.66. Below 50: not passing

- Final grade is based on *average* of two exams, plus labs, participation, and final project grades.
- If your grade is below 60, please come talk to me.
- Everyone is encouraged to revisit anything where they missed many points.
- Any questions about your exam grade: bring to me within one week.

Useful string methods

```
txt = "50800"

#returns True if all the characters are digits, otherwise False
print(txt.isdigit())


print(txt.isalpha()) #returns True if all characters are alphabet letters
(a-z)

s = "Hello!"

print(s.lower()) #returns a string where all characters are lower case,
                 #symbols and numbers are ignored
```

Modules and Libraries

We've used "import" in several ways, to get things into your program from:

- a standard system library, e.g., math (for sqrt, etc.)
- a non-standard thing from Haverford computers, e.g. Logic (precondition, etc.)
- something else from your project (e.g., as done in A_file_for_debugging.py)

Read from & Write to Files

```
open("Downloads/CS105/demofile.txt") # returns a file object

f = open("demofile.txt", "r")        # "r" parameter for reading file
print(f.read())                      # reads content of entire file
print(f.readline())                  # reads one line
f.close()                            # close the file when you are done with
it

f = open("demofile.txt", "a")        # append to the end of the file
f.write("Now the file has more content!")
f.close()

f = open("demofile.txt", "w")        # overwrite any existing content
f.write("Whoops! I have deleted the content!")
f.close()
```

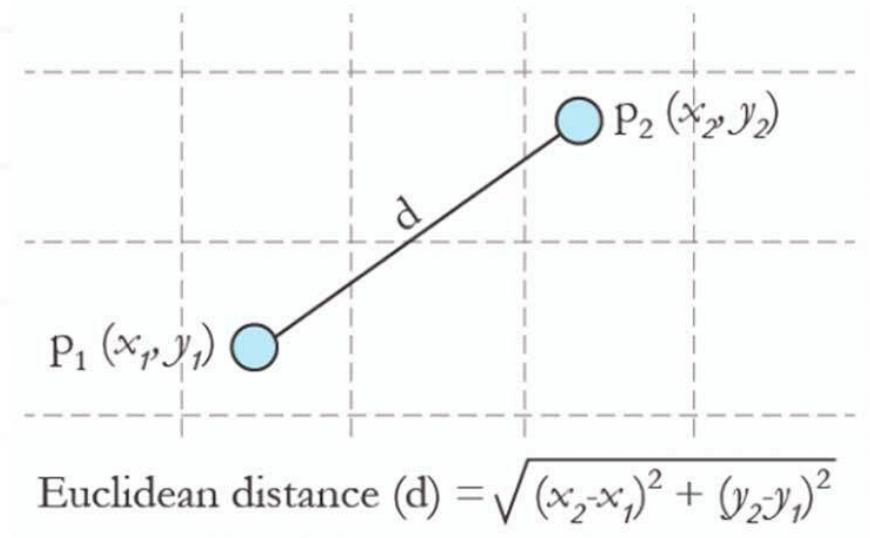
Distance Metrics

Hamming distance: compare 2 strings

Count the number of characters that differ.

Euclidean distance:

Straight line distance between 2 points.



Dictionary Practice

Write a python function sum_values() that takes *one* dictionary of string/float data and two strings that should be keys in the dictionary, and returns the sum of their associated values.

For example, if we create a dictionary of fruits and their masses, we could get the total weight of a watermelon and a grape like this:

```
>>> from typing import Dict
>>> fruit_mass_in_grams: Dict[str, float] = {'grape':5.1, 'apple':102,
'watermelon': 5440}
>>> sum_values(fruit_mass_in_grams, 'watermelon', 'grape')
5445.1
```