שילוב C++ עם פייתון

שפת C++ היא שפה מאד יעילה ומהירה, אבל גם לא-כל-כך נוחה למתכנת (כמו שודאי שמתם לב אחרי 5 מטלות..)

לעומת זאת, שפת פייתון (**Python**) היא שפה חדשה, מאד קלה ונוחה לתיכנות (תיכוניסטים לומדים אותה לבד תוך שבוע), אבל מאד לא יעילה – נראה בהמשך השיעור דוגמה לתוכנית שרצה בפייתון פי 40 (ארבעים!) מתוכנית זהה בשפת C++.

במקרים רבים נרצה לשלב את היעילות של C++ עם הנוחות של פייתון. לדוגמה, ביישומים של למידת מכונה, אנחנו רוצים מצד אחד שהחישובים יתבצעו מאד מהר – אחרת הם ייקחו שנים רבות (במיוחד בלמידה עמוקה). מצד שני, אנחנו רוצים מערכת נוחה לתיכנות, כך שגם אנשים מחוץ למדעי המחשב (כגון פרופסורים לסוציולוגיה, ספרות וכד') יוכלו להשתמש בה לצרכיהם.

הפתרון הוא לשלב את C++ עם פייתון. ישנן דרכים רבות לשלב, אנחנו נלמד דרך אחת חדשה ונוחה במיוחד – cppyy.

היכרות עם פייתון

השלב הראשון הוא להתקין על המחשב שלכם את שפת פייתון (גירסה 3 ומעלה), וללמוד לעבוד איתה. יש כמה דוגמאות בסיסיות ביותר בתיקיה 1. שימו לב איך מגדירים פונקציות ומשפטי תנאי, ו איך מייבאים חבילות (import).

היכרות עם cppyy

השלב השני הוא להתקין את cppyy. ניתן להתקין אותה דרך מנהל-החבילות של פייתון, למשל:

```
pip3 install cppyy
```

לפרטים נוספים, וכן אם אתם נתקלים בבעיות בהתקנה, אנא קיראו בתיעוד של cppyy.

אחרי שהתקנתם את החבילה, היכנסו לדוגמאות הקוד בתיקיה 2. הקובץ hello.py הוא תוכנית פייתון שגם מגדירה פונקציה ב-C++ בעזרת הפקודה: cppyy.cppdef. לאחר שהגדרנו פונקציה בשם say_hello, היא הופכת להיות שיטה של העצם הגלובלי cppyy.gbl, ואפשר פשוט לקרוא לה כמו לכל שיטה אחרת בפייתון: cppyy.gbl.say_hello().

דרך נוספת לייבא קוד C++ לתוך פייתון היא הפקודה cppyy.include – ראו בקובץ include.py. התוצאה זהה – הקוד המוכלל (מתוך hello.hpp) הופך להיות שיטה של העצם הגלובלי cppyy.gbl.

פרמוטציות

עכשיו נראה דוגמה שבה המעבר ל-C++ חוסך הרבה מאד זמן. נניח שאנחנו רוצים לפתור את בעיית הסוכן הנוסע – הסוכן שצריך לעבור בכל הערים, כך שהמרחק הכולל שהוא עובר יהיה הקצר ביותר. כידוע לכם, זו בעיה קשה מאד. הפתרון הכי פשוט (והכי לא יעיל) לבעיה, הוא פשוט לעבור על כל

הפרמוטציות של כל הערים, לחשב את המרחק הכולל עבור כל פרמוטציה, ולבחור את הפרמוטציה עם המרחק הכולל הקצר ביותר.

בתור הקדמה לבעיה, נפתור קודם בעיה אחרת – לספור את כל הפרמוטציות (כמובן, אנחנו יודעים שמספר הפרמוטציות על n איברים הוא n!, אבל לצורך הדוגמה נניח שאנחנו רוצים לעבור עליהן אחת אחת ולספור אותן).

בקובץ count_permutations.py יש קוד שעושה את זה בשפת פייתון – מאד פשוט קצר וקריא.

בקובץ count_permutations.cpp יש קוד שעושה את זה בשפת ++C - תוך שימוש באלגוריתמים של הספריה התקנית שלמדנו בשיעור קודם.

בקובץ count_compare.py יש קוד שמייבא את count_permutations.cpp בעזרת cppyy.include, ומריץ את שתי הפונקציות – זאת שכתובה בפייתון וזאת שכתובה ב ++C – על אותו קלט.

התוצאות מדהימות – הקוד ב ++C רץ פי 40 יותר מהר, ואפשר לקרוא לו מתוך פייתון באותה רמת נוחות כמו לקוד פייתון מקורי.

תופעה דומה קורה בבעיית הסוכן-הנוסע עצמה – ראו בקבצים traveling_salesman.py, traveling_salesman.cpp, traveling_compare.py. שימו לב לאופן שבו אנחנו מעבירים קלט ל-++C: אנחנו מעבירים וקטור של וקטורים בפייתון, והוא מתורגם אוטומטית לוקטור של וקטורים ב-++C.

השגנו את המטרה – שילבנו נוחות עם מהירות.

סיכום

בדרך-כלל, כשמתחילים לכתוב קוד לצורך הדגמה או בדיקת רעיון חדש, הכי נוח לכתוב אותו בפייתון – התיכנות מאד קל ומהיר. בהמשך, כשהמערכת גדלה והופכת להיות "מבצעית", בודקים את המקומות הקריטיים שבהם מתבזבז זמן רב במיוחד, מתרגמים אותם לשפת ++C, ומשלבים בין השפות בעזרת cppyy או כלי אחר כלשהו.

היתרון שלכם כבוגרי מדעי-המחשב (לעומת "סתם" קורס תיכנות) הוא, שאתם מכירים הרבה שפות והרבה טכנולוגיות שונות, יודעים מה היתרונות והחסרונות של כל שפה, יודעים לבחור את השפה המתאימה למשימה המתאימה, ויודעים גם לשלב בין שפות לפי הצורך.

מקורות

- התיעוד של cppyy: https://cppyy.readthedocs.io/en/latest/

סיכם: אראל סגל-הלוי.