שכפול פונקציונליות בתרגיל גרפ, ואיך להמנע ממנו

מבנה קוד בעייתי - אל תעשו את זה:

- כיצד נבדוק התאמה?
- אם יש שימוש בסוויץ' E, קוראים לפונקציה א'.
- אם יש שימוש בסוויץ' X אך לא ב-E, קוראים לפונקציה ב'.
- אחרת, קוראים ל-strstr.
-
- זה שכפול פונקציונליות!

למה זה שכפול פונקציונליות

- כי פעולה לוגית אחת - בדיקה האם יש התאמה בשורה - מתבצעת בכמה דרכים שונות.
- נניח שרוצים להוסיף תמיכה בפורמט תווים יותר מורכב בשם unicode.
- עכשיו אנחנו צריכים לעבור על שלושת הדרכים בהן בדיקת ההתאמה מתבצעת, ולהוסיף את התמיכה בכל אחת מהדרכים.
- כמו כן, הדרך הראשונה (שמטפלת ב-E-) מספיק כללית כדי לטפל במקרים האחרים.
- כתוצאה מכך, אנחנו מפסידים את היתרונות של חלוקה לרכיבים עליהם דיברנו:
  - כל באג צריך לתקן בכמה מקומות, במקום במקום בודד.
  - (כפי שראינו) כל שינוי צריך לבצע בכמה מקומות, במקום במקום בודד.

מה כן לעשות

- לכתוב פונקציה אחת שהיא מספיק כללית כדי לבדוק האם יש התאמה בשורה, בכל המצבים (סוויצ'ים) האפשריים.

מחיקת קוד

- האם השתמשתם בעיקר ב-strstr ו/או strcmp עבור בטא1 ובטא2? (הגיוני)
- **אזי, זה אומר שאתם תצטרכו למחוק קוד שכתבתם.**
  - האם אתם מעדיפים לכתוב מראש את הקוד כך שיעבור את הגירסה הסופית? (זה בסדר מבחינת ההנחיות...)
  - פיגומים
- למחוק קוד שלא צריך - נתפס בתור דבר מבורך בעולם התוכנה.

Case Study: BoringSSL

- Background: Forking.
- BoringSSL: Encryption Library forked by Google (from OpenSSL)
- https://www.imperialviolet.org/2015/10/17/boringssl.html
- “avoids some duplicated code”
- “allowing both projects to shed some code”
- “large amounts of OpenSSL could simply be discarded”
- Lines of Code: 460k -> 200k

- (unrelated) “In C, any malloc call may fail. OpenSSL attempts to handle this, but such code is error-prone and rarely tested.”

Case Study: DROWN

- “Support for SSLv2, X, Y and Z has all been dropped.”
  - (In BoringSSL, not in OpenSSL)
- In OpenSSL: SSLv2 code lives on (in 2015)...
- But is disabled, because of configuration options.
- Surprise: These options are actually ignored.
- SSLv2 code is from 1998 or earlier.
- So in practice, SSLv2 code is active, reachable.
- End result: Huge vulnerability in 33% of Internet
  - (disclaimer: Yours truly was a coauthor)
- Cf. similar vuln. that could be prevented by shedding code: FREAK