

אלגוריתמים, סמסטר ב' תשפ"א המכללה האקדמית של תל-אביב-יפו

פרויקט תכנותי

תאריך הגשה: יום רביעי ה- 12.5.21 עד שעה 23:59.

הנחיות כלליות

1. הפרויקט ניתן להגשה בבודדים או בזוגות, אך לא בקבוצות גדולות יותר. מומלץ להגיש את הפרויקט בזוגות.
2. איחור בהגשה יאושר רק במקרה של מילואים, מחלה ממושכת או לידה, וגם זאת רק בתנאי שהפנייה למרצה בנושא נעשתה לפני מועד ההגשה המקורי של הפרויקט.

מטרת הפרויקט

עליכם לכתוב תכנית בשפת C++ (ובה בלבד. בחירת השפה אינה נתונה לשיקול דעת הסטודנטים), שפותרת בכמה דרכים את **בעיית המרחקים המינימליים ממוצא יחיד ליעד יחיד** בגרפים מכוונים, לפי כמה אלגוריתמים שלמדנו בקורס, וכן לחשב את זמן הריצה של הפתרונות השונים.

השלבים לביצוע הפרויקט

- א. קראו תחילה היטב את ההנחיות של הפרויקט והבינו מה נדרש בדיוק.
- ב. תכננו את ה-design של התוכנית שלכם: בחרו אילו מחלקות תממשו, החליטו על data members מתאימים ועל methods רלוונטיים לכל מחלקה.
- ג. כתבו מימוש מלא לכל המחלקות שייעשה בהן שימוש במסגרת התוכנית.
- ד. ממשו את האלגוריתמים הנדרשים ואת פונקציית ה-main של התוכנית.

תיאור מפורט של מטרת הפרויקט:

מטרת הפרויקט היא לממש מספר פתרונות ל**בעיית המרחקים המינימליים ממוצא יחיד ליעד יחיד**, לחקור ולנתח את ההבדלים בין הפתרונות השונים.

המרחקים המינימליים ממקור יחיד יחושבו על ידי שני אלגוריתמים מרכזיים:

1. אלגוריתם דיקסטרה כאשר תור הקדימויות ממומש על ידי ערימת מינימום.

2. אלגוריתם דיקסטרס כאשר תור הקדימויות ממומש על ידי מערך רגיל ללא סדר כלשהו בין האיברים.
3. אלגוריתם בלמן-פורד.

לכל אחת משלוש האפשרויות האלה יש להריץ את האלגוריתם פעם אחת כאשר הגרף ממומש על ידי רשימת שכנויות ופעם אחת כאשר הגרף ממומש על ידי מטריצת שכנויות. כלומר סה"כ יחושבו המרחקים המינימליים 6 פעמים. בנוסף עליכם לכתוב לקובץ את זמן הריצה לכל אחת מהאפשרויות האלה.

תיאור התכנית הראשית :

פונקציית main של התכנית תקבל שני ארגומנטים שהם שמות קבצים ותבצע:

1. קליטת נתוני הגרף מקובץ טקסט ששמו נמצא בארגומנט הראשון (`argv[1]`) ויכיל את הקלט הבא:
 - 1.1 מספר שלם, n , שמסמן את מספר הקדקודים בגרף.
 - 1.2 מספר שלם המייצג את שם קדקוד המוצא s ומספר שלם t המייצג את קדקוד היעד.
 - 1.3 שלשות של מספרים, כל שלשה בשורה נפרדת, כאשר השלשות מייצגות שמות של שני קדקודים i, j (כל קדקוד מיוצג על ידי מספר בין 1 ל- n) ואת משקל הקשת מ- i ל- j .
 - בעזרת נתונים אלה התוכנית תבנה גרף, פעם אחת כמטריצת שכנויות ופעם אחת כרשימות שכנים.
 2. תריץ את האלגוריתמים המתוארים למעלה (סה"כ 6 הרצות).
 3. תדפיס למסך את משקל המסלול המינימלי מ- s ל- t בכל אחד מהמקרים ותדפיס לקובץ ששמו נתון בארגומנט השני (`argv[2]`) את זמן הריצה בכל אחד מ- 6 המקרים.
- פורמט הפלט יהיה (בהנחה שמשקל המק"ב מ- s ל- t הוא $\langle x \rangle$):

Adjacency Dijkstra heap $\langle x \rangle$
Adjacency Dijkstra array $\langle x \rangle$
Adjacency Bellman Ford $\langle x \rangle$
Matrix Dijkstra heap $\langle x \rangle$
Matrix Dijkstra array $\langle x \rangle$
Matrix Bellman Ford $\langle x \rangle$

כך למשל קלט תקין יראה כך:

```
6
1
6
1 2 16
1 3 13
2 3 10
3 2 4
2 4 12
4 3 9
3 5 14
5 4 7
5 6 4
4 6 20
```

במקרה זה הפלט יהיה:

```
Adjacency Dijkstra heap 31
Adjacency Dijkstra array 31
Adjacency Bellman Ford 31
Matrix Dijkstra heap 31
Matrix Dijkstra array 31
Matrix Bellman Ford 31
```

מבני הנתונים

לצורך ביצוע הפרויקט, הנכם נדרשים לממש את המחלקות הבאות ללא שימוש ב-STL:

1. גרף מכוון פשוט עם משקלים שממומש על ידי מטריצת שכנויות:

הגרף ימומש כמטריצת שכנויות, אשר תכיל את משקלי הקשתות של הגרף. כלומר אם A היא המטריצה, אז $A[u][v]$ = משקל הקשת מ- u ל- v .

פעולות בסיסיות:

- MakeEmptyGraph(n)** – יצירת גרף ריק מקשתות עם n קדקודים.
- IsAdjacent(u,v)** – מחזיר כן אם הקשת (u,v) שייכת לגרף, ואחרת לא.
- GetAdjList(u)** – החזרת רשימה מקושרת של השכנים של קדקוד u .
- AddEdge(u,v,c)** – הוספת קשת (u,v) בעלת משקל c .
- RemoveEdge(u,v)** – הסרת הקשת (u,v) מהגרף.

2. גרף מכוון פשוט עם משקלים שממומש על ידי רשימות שכנויות:

הגרף ימומש כרשימות שכנויות, כאשר כל קדקוד ברשימת שכנויות יכיל בנוסף למצביע לקדקוד הבא ברשימה גם שדה של משקל הקשת המתאימה.

3. תור קדימויות מינימום שממומש על ידי ערימת מינימום.

תור הקדימויות ימומש בעזרת ערימת מינימום בינארית שתישמר במערך, כפי שנלמד בקורס מבני נתונים.

כל איבר בתור יהיה זוג מהצורה (data, key) כאשר העדיפות נקבעת על פי ה-key.

פעולות בסיסיות:

Build – בניית ערימה מתוך מערך על ידי שימוש באלגוריתם של פלוד.
DeleteMin – שליפת האיבר עם המפתח המינימלי מהערימה ותיקון הערימה.
IsEmpty – בדיקה האם הערימה ריקה.
DecreaseKey(place, newKey) – הקטנת ערך המפתח (עדיפות) של האיבר שנמצא במקום place בערימה לערך newKey (וכמובן תיקון הערימה לפי הצורך).

4. תור קדימויות מינימום שממומש על ידי מערך רגיל פלוס משתנה ששומר את האיבר עם המפתח המינימלי.

עליכם לממש לכל אחד מטיפוסי הנתונים המופשטים האלה את הפעולות הבסיסיות המפורטות לעיל. מעבר לכך אתם יכולים להוסיף פעולות ו- data members נוספים, כראות עיניכם.

כמו כן עליכם לממש כמובן כל מחלקה אחרת שלה תזדקקו במהלך התכנית.

בדיקת שגיאות

הקפידו לבדוק שגיאות אפשריות בקלט (למשל לבדוק שהגרף פשוט, שאין משקלים שליליים וכך הלאה). במקרה של שגיאה, כתבו הודעת שגיאה למסך invalid input וצאו מהתכנית באמצעות הפונקציה exit(1).

(יש לבצע `#include <stdlib.h>` על מנת להשתמש בה).

הנחיות הגשה

יש להגיש במערכת mama במקום המיועד להגשה את הקבצים הבאים:

1. קובץ readme שיכיל את כל פרטי ההגשה הבאים:

כותרת – פרויקט בקורס אלגוריתם.

שורה מתחת - שמות המגשים, מספרי ת.ז. שלהם ומספר הקבוצה של כל אחד מהם (מותר להגיש עם בן זוג מקבוצה אחרת).

שימו לב: קובץ טקסט פשוט – לא word.

2. כל קבצי הקוד בסיומות .cpp ו- .h.

3. קובץ ה- solution של visual studio

4. רק אחד מבני הזוג יגיש את הפרויקט.

שימו לב! הגשה שאינה בפורמט הנדרש תידחה אוטומטית.

הערות הכרחיות נוספות (שימו לב, למרות שזה קורס באלגוריתמים התכנית

צריכה להיות כתובה לפי כל הכללים של תכנות נכון!)

- הקפידו על תיעוד, שמות בעלי משמעות למשתנים, מודולאריות וכל מה שנדרש מתכנות נכון. בתיעוד בראש התוכנית כתבו גם הוראות הפעלה מדויקות וברורות.
- הקפידו על חלוקה נכונה לקבצים (קבצי cpp וקבצי h לכל מחלקה).
- תכנתו Object Oriented והימנעו מאלמנטים מיותרים של קוד פרוצדוראלי.
- הקפידו לשחרר את כל הזיכרון אשר הקציתם דינמית לאחר שהשתמשתם בו.
- במקרה של תקלה בריצת התוכנית (מסיבה כלשהי), עליה לדווח זאת למשתמש טרם סיימה לרוץ.
- בדקו את תכניותיכם על קלטים רבים ככל האפשר - כולל קלטים חוקיים ולא חוקיים, וזאת בנוסף לקלטים לדוגמה שהוכנו עבורכם במאמא.
- **תנאי הכרחי (אך כמובן לא מספיק) לקבלת ציון עובר על הפרויקט**, הוא שהקוד יעבור **קומפילציה בויז'ואל סטודיו 2019**. ציונו של פרויקט אשר אינו עובר קומפילציה יהיה 0.

צוות אלגוריתמים, סמסטר ב' תשפ"א.