27

נושא 6 - BFS (Breadth-First Search)

א) הגדרה

אלגוריתם BFS בדומה לאלגוריתם DFS מקבל גרף G לא ממושקל וצומת מקור כלשהי S וסורק את הגרף הזה מצומת זו. אלא שבניגוד לאלגוריתם DFS הסורק לעומק (מתקדם במסלול עד שנתקע וחוזר), אלגוריתם BFS סורק לרוחב. כלומר קודם עובר בכל הצמתים שבמרחק קשת אחת מ-S, ואז כל הצמתים במרחק שתי קשתות מ-S, וכן הלאה.

הבדל נוסף הוא במטרת האלגוריתם. מטרת אלגוריתם DFS הוא לסרוק את כל צמתי הגרף ולקבל עליהם מידע לשימוש אלגוריתמים אחרים כמו שראינו. לעומת זאת, מטרות אלגוריתם BFS הם:

(1) למצוא את כל הצמתים שניתן להגיע אליהם בגרף מ-S. רכיב הקשירות של S.

(2) למצוא את המסלול הקצר ביותר מ-S לכל צומת אחרת שניתן להגיע אליה מ-S.

המסלול הקצר ביותר הוא המסלול עם הכי מעט קשתות מ-S לצומת.

כדי לעשות פעולות אלו האלגוריתם מתחזק שני מערכים:

d[v] - שומר את מספר הקשתות מ-S ל-V. גודל זה הוא גודל המרחק הקצר ביותר מ-S ל-V.

π[v] - שומר את הצומת שהובילה לגילוי של V. צומת זו היא הצומת האחרונה במסלול מ-S ל-V.

בנוסף האלגוריתם שומר לכל צומת צבע כדי לעקוב אחר הצמתים שהוא כבר ביקר.

לבן - בתחילה כל הצמתים לבנים, צבע זה מסמל שעוד לא ביקרנו בצומת.

אפור - מסמל שביקרנו בצומת אך עוד לא עברנו בכל השכנים של צומת זו.

שחור - מסמל שביקרנו בצומת ועברנו על כל השכנים שלה.

האלגוריתם משתמש במבנה נתונים מסוג תור (Queue) כדי לקבוע איזה הצמתים הבאים בהם יבקר.

ב) פסאודו-קוד

```
BFS(G,S) {
    For each u∈V {
        color[u] = white , d[u] = ∞ , π[u] = null
    }
    color[s] = gray , d[s] = 0 , π[s] = null
    Q = {s}          // אתחול תור Q להכיל רק את s
    while (Q ≠ ∅) {
        u = Q.dequeue     // u שולף את האיבר הראשון בתור
        For each v∈ adj[u] {
            If (color[v] = white)
                d[v] = d[u] + 1 , π[v] = u
                Q.enqueue(v)     // מכניס את v לסוף התור
        }   }
        color[u] = black     // לאחר שעברנו על כל השכנים נצבע בשחור
}   }
```

2ג

ג) זמן ריצה

אלגוריתם BFS במקרה הגרוע שבו הגרף קשיר יעבור על כל צמתי הגרף. ועבור כל צומת עובר על כל הקשתות שיוצאים מצומת זו. לכן הסיבוכיות היא $O(m+n)$.

ד) מסלול קצר ביותר

כמו שדיינו. באמצעות אלגוריתם BFS ניתן למצוא את המסלול הקצר ביותר מ-s לכל צומת $v \in V$. באלגוריתם זה, מכיוון שהגרף אינו ממושקל נגדיר מסלול הכי קצר כמסלול עם הכי מעט קשתות. נסמן גודל המרחק הכי קטן מ-s ל-v כך: $\delta(s,v)$.

כאשר אנו מגלים צומת v ומעדכנים את השדות שלו, מתקיים $d[v]=\delta(s,v)$, כלומר כבר בעת הגילוי מצאנו את אורך המסלול הכי קצר ושמרנו אותו ב- $d[v]$.

כדי למצוא את המסלול עצמו של הצמתים נלך מ-v אחורה אל הצומת ב- $\pi[v]$, בצומת זו שנסמנה u שוב נלך אל הצומת ב- $\pi[u]$, וכן הלאה עד שנגיע ל-s. כל צמתים אלו הם המסלול הקצר ביותר מ-s ל-v, ומספרם $\delta(s,v)+1$.

ה) עץ BFS

באמצעות כל המסלולים הקצרים ביותר מ-s לכל צומת אחרת בגרף, ניתן לבנות עץ BFS שבו s הוא השורש וכל שאר הצמתים בעץ הם צמתים שניתן להגיע אליהם מ-s. כאשר הצמתים בגובה 1 הם הצמתים שהמרחק הקצר ביותר מ-s אליהם הוא קשת אחת, בגובה 2 שתי קשתות וכן הלאה. צומת v שעבורו $\pi[v]=null$ לא ניתן להגיע מ-s אליו.

ו) הוכחת האלגוריתם

<u>טענת עזר:</u> אם צומת v נכנס לתור לפני צומת u, אזי $d[v] \le d[u]$.

<u>טענה:</u> נוכיח כי בסוף האלגוריתם עבור כל צומת $v \in V$ מתקיים $d[v]=\delta(s,v)$.

הוכחה: נניח בשלילה כי הטענה אינה נכונה, וכי v הצומת הראשון שבו $d[v]>\delta(s,v)$.* ונניח כי u היא הצומת שלפני v במסלול מ-s ל-v. מהמינימליות של v נובע כי $d[u]=\delta(s,u)$.

*לא מתקיים $d[v]<\delta(s,v)$ כי עוברים באיזו דרך על הקשתות בגרף.

לפי ההנחה בשלילה מתקיים: $d[v]>\delta(s,v)=\delta(s,u)+1=d[u]+1$. נתבונן בכל האפשרויות ל-v כאשר u נמחק מהתור:

v לבן - לא יכול להיות כי יש קשת מ-u ל-v והאלגוריתם היה מגלה את v לפני ש-u נמחק.

v אפור - נניח כי צומת w גילה את v לפני u ו- $d[v]=d[w]+1$. אמנם אז w נמחק לפני u ולכן לפי טענת עזר נקבל כי $d[v] \le d[u]+1$ וזוהי סתירה להנחה שלנו לעיל.

v שחור - ואז שוב נקבל לפי טענת עזר $d[v] \le d[u]$ בסתירה להנחה לעיל.