

## שאלה 2 סעיף א':

אם לכל החבילות הנכנסות יש אותם ערכים, אך הערכים של הסלאקים שלהם שונים עדיף להשתמש באלגוריתם EDF.

אלגוריתם EDF משדר את החבילה עם ערך הסלאק הנמוך ביותר. במהלך ריצת האלגוריתם, כאשר התור מלא, האלגוריתם משווה בין הערכים של הסלאקים של החבילות שכבר נמצאות בתור כאשר הוא מלא לבין ערך הסלאק של החבילה שרוצה להכנס לתור. אם החבילה שרוצה להתקבל היא עם ערך סלאק גבוה יותר מהחבילה עם ערך הסלאק הנמוך ביותר בתור, היא מתקבלת לתור ומחליפה את זו שהשוואנו אותה בה. אם החבילה שרוצה להתקבל היא עם ערך סלאק נמוך יותר מהחבילה עם ערך הסלאק הנמוך ביותר בתור, היא נזרקת.

ח' - בכל ירידת שורה בקריאה של הקובץ, האלגוריתם מוריד 1 מערכי הסלאק של כל חבילה בתור ומשדר את החבילה עם ערך הסלאק הנמוך ביותר בתור וסוכם את הvalue שלה לvalue הכללי של כל החבילות ששודרו.

EOF - כאשר כל החבילות נשלחו (כל החבילות נקראו מהקובץ), האלגוריתם מוריד מהחבילות בתור את הערכים של הסלאקים ב-1, החבילות עם ערך סלאק 0 נזרקות. האלגוריתם בוחר בכל רוטציה איזו חבילה הוא משדר, הוא בוחר לשדר את החבילה עם ערך הסלאק הנמוך ביותר, מוציא אותה מהתור וסוכם את הvalue שלה לvalue הכללי של כל החבילות ששודרו, עד שהתור מתרוקן.

בגלל שהערך של כל חבילה הוא קבוע לפי ההנחה, אז הערך הסופי נקבע להיות מספר החבילות אשר שודרו כפול הערך הקבוע הזה, לפי מה שצויין למעלה, לא יוכל להיות מצב כי אלגוריתם EDF יזרוק חבילה (כלומר הסלאק שלה יגיע ל0) בעוד שאלגוריתם BD כן, ולכן כמות החבילות אשר יתקבלו באלגוריתם EDF הינה גדולה שווה לכמות החבילות אשר ישודרו באלגוריתם BD, ולכן זה גורר לכך שהערך הסופי (total value) של אלגוריתם EDF יהיה גדול שווה לערך הסופי של אלגוריתם BD.

הערה: הערך הסופי יכול להיות גדול כי אלגוריתם BD יזרוק חבילה (ערך הסלאק שלה יגיע ל0) בעוד שאלגוריתם EDF יבצע החלפה לפני שערך הסלאק של החבילה יגיע לאפס ולא יזרוק את החבילה, האלגוריתם ישדר אותה ויסכום אותה בערך הסופי.

## שאלה 2 סעיף ב':

אם לכל החבילות יש את אותם ערכים של סלאקים, אך הערכים שלהם שונים עדיף להשתמש באלגוריתם BD.

אלגוריתם BD משדר את החבילה עם הערך הגבוה ביותר. במהלך ריצת האלגוריתם, כאשר התור מלא, האלגוריתם משווה בין הערכים של החבילות שכבר נמצאות בתור כאשר הוא מלא לבין ערך החבילה שרוצה להכנס לתור. אם החבילה שרוצה להתקבל היא בעלת ערך גבוה יותר מהחבילה בעלת הערך הנמוך ביותר בתור, היא מתקבלת לתור ומחליפה את זו שהשוואנו אותה בה. אם החבילה שרוצה להתקבל היא בעלת ערך נמוך יותר מהחבילה עם הערך הנמוך ביותר בתור, היא נזרקת.

ח' - בכל ירידת שורה בקריאה של הקובץ, האלגוריתם מוריד 1 מערכי הסלאק של כל חבילה בתור ומשדר את החבילה עם ערך הגבוה ביותר בתור וסוכם את value שלה לvalue הכללי של כל החבילות ששודרו.

EOF - כאשר כל החבילות נשלחו (כל החבילות נקראו מהקובץ), האלגוריתם מוריד מהחבילות בתור את הערכים של הסלאקים ב-1, החבילות עם ערך סלאק 0 מזרקות.  
האלגוריתם בוחר כל רוטציה איזו חבילה הוא משדר, הוא בוחר לשדר את החבילה בעלת הערך הגבוה ביותר, מוציא אותה מהתור וסוכם את value שלה לvalue הכללי של כל החבילות ששודרו, עד שהתור מתרוקן.

נשים לב כי אלגוריתם BD מחליף (במידה והתור מלא) את החבילות בעלות הערך הנמוך ביותר בחבילה הנכנסת (במידה והערך של החבילה הנכנסת הוא גדול יותר מהערך של החבילה בעלת הערך הנמוך ביותר בתור), בגלל שאלגוריתם EDF בוחר את החבילה בעלת הסלאק הנמוך ביותר ובגלל שהסלאק הוא זהה לכל החבילות לפי ההנחה, אז הוא יבחר את אחת מבין החבילות הללו ללא הסתכלות על הערך שלהן, אז הערך המצטבר בתור לא בהכרח עולה, בעוד שהערך המצטבר בתור באלגוריתם BD יכול רק לעלות.  
בנוסף נשים לב שבאלגוריתם BD כאשר יורדים שורה אז קודם כל מורידים באחד את הערך של הסלאק של כל החבילות בתור ולאחר מכן מקבלים את החבילה בעלת הערך הגבוהה ביותר, בעוד שב-EDF מקבלים את זאת עם הסלאק ההנמוך ביותר.  
לכן בגלל שהסלאק זהה, נרצה לקבל את החבילה בעלת הערך הגדול ביותר ולא את החבילה בעלת הסלאק הנמוך ביותר בין החבילות בתור, היות וכל החבילות בעלות אותו סלאק נעלמות ביחד (כאשר הן מגיעות ל-0).

## שאלה 2 סעיף ג':

במקרה הכללי שחבילות נכנסות עם ערכים שונים ועם ערכים שונים של סלאקים, אראה מתי עדיף להשתמש באלגוריתם EDF ומתי באלגוריתם BD.

\*מושגים אשר מסומנים בכוכבית יוסברו בהמשך.  
האלגוריתמים יצטרכו לקרוא קריאה מקדימה של הקובץ כדי לדעת באיזה אלגוריתם עדיף להשתמש.

### **- החלפה כאשר התור מלא:**

- כאשר יש לנו חבילות אשר \***ילכו לאיבוד** יש עדיפות להשתמש באלגוריתם EDF מפני שחבילות עם ערך סלאק נמוך בתור ילכו לאיבוד ואם ישנה חבילה שרוצה להכנס לתור והיא בעלת ערך סלאק גבוה יותר מהחבילה בעלת הסלאק הנמוך ביותר בתור אז עדיף להחליף בניהן כדי לקבל את החבילה ולא לאבד אותה.

- כאשר אין לנו חבילות אשר \***ילכו לאיבוד** יש עדיפות להשתמש באלגוריתם BD מפני שחבילות עם ערכים גדולים יותר יעלו את הסכימה הסופית של החבילות וזוהי המטרה היות ואני רוצה למקסם ערכים, ולכן האלגוריתם יחליף את החבילה עם הערך הנמוך ביותר בתור לחבילה שרוצה להכנס לתור והיא בעלת ערך גבוה יותר מהחבילה שהאלגוריתם ירצה להוציא.

### **- מה נקבל בירדת שורה: ח'**

- כאשר יש לנו חבילות אשר \***ילכו לאיבוד** יש עדיפות להשתמש באלגוריתם EDF מפני שחבילות עם ערך סלאק נמוך בתור ילכו לאיבוד כאשר נוריד 1 מערך הסלאק שלהן והשאיפה היא לשמר אותן ולכן נקבל אותן לפני שיאבדו.  
- כאשר אין לנו חבילות אשר \***ילכו לאיבוד** יש עדיפות להשתמש באלגוריתם BD מפני שחבילות עם ערכים

גדולים יותר יעלו את הסכימה הסופית של החבילות והערך שלהן מספיק גבוה ולכן אין את ה"חשש" שהחבילות הנ"ל יאבדו.

#### **- מה נקבל בסוף הקובץ: EOF**

- כאשר יש לנו חבילות אשר \***ילכו לאיבוד** יש עדיפות להשתמש באלגוריתם EDF מפני שבכל איטרציה בתור הערך של הסלאק של החבילות יורד ב-1 ובחרים חבילה אחת וסוכמים את הערך שלה. מס איטרציות = גודל התור. ולכן יש עדיפות לקבל את החבילה עם הערך הסלאק הנמוך ביותר לפני שנאבד אותה.

- כאשר אין לנו חבילות אשר \***ילכו לאיבוד** יש עדיפות להשתמש באלגוריתם BD מפני שחבילות בתור לא יאבדו בזמן האיטרציות ונרצה לסכום קודם כל את החבילות עם הערך הגדול ביותר כדי שהסכימה הסופית שלנו תגדל כמה שיותר.

**ילכו לאיבוד:** הסלאק של החבילה קטן שווה מכמות השורות בקובץ+הערך שלה נמוך יחסית לחבילות האחרות (כלומר לא תבחר בירידת שורות) ולכן כאשר נגיע ל-EOF היא תיעלם בגלל שכל פעם שאנחנו מקבלים חבילה בכל שורה אנחנו מורידים את הסלאק ב-1.

#### **שאלה 2 סעיף ד':**

חשבתי על אלגוריתם אשר בעצם משלב את 2 האלגוריתמים שהיו בסעיף א' ו-ב', בעצם חשבתי איך ליצור אלגוריתם שיועד איזו חבילה להחליף במידה והתור מלא ועדיין יש חבילות שרוצות להכנס לתור (אם התור לא מלא אז נכניס את החבילות עד שיתמלא התור) ובדומה איזו חבילה לקבל כאשר ישנה ירידת שורה בקובץ או כאשר סיימתי לקרוא את כל הקובץ שמתוכו אני קוראת את החבילות.

האלגוריתמים אשר הוצגו בסעיף א' ו-ב', מורידים 1 מערך הסלאק בכל ירידת שורה בקובץ ואז מבצעים קבלה של חבילה ונכ"ל כאשר מגיעים ל-EOF ובכל איטרציה מקבלים חבילה עד שהתור מתרוקן. ולכן באלגוריתם החדש אגדיר איזו חבילה מתחלפת כאשר התור מלא ויש עוד חבילות שרוצות להכנס, איזו חבילה מתקבלת בכל ירידת שורה ואיזו חבילה מתקבלת בכל איטרציה לאחר שמגיעים ל-EOF ועד שהתור מתרוקן.

#### **האלגוריתם פועל באופן הבא:**

\*מושגים אשר מסומנים בכוכבית יוסברו בהמשך.

#### **- החלפה כאשר התור מלא:**

נסתכל על קבוצת כל החבילות S אשר \***ילכו לאיבוד**, מתוכן האלגוריתם יחליט איזה מהן שווה "להציל". אם אחת החבילות בקבוצה S הולכת לאיבוד באיטרציה הקרובה, כלומר הסלאק שלה שווה בדיוק ל-1 אז האלגוריתם ירצה להחליף אותה בלי קשר לערך שלה. אחרת, יבחר את החבילה בעלת הערך הנמוך ביותר ויחליף אותה אם \***שווה לו**.

אם S היא קבוצה ריקה, האלגוריתם יבחר את החבילה בעלת הערך הקטן ביותר בתור ויחליף אותה בחבילה שרוצה להכנס לתור אם היא לא חבילה שתלך לאיבוד (\***ילכו לאיבוד**) והערך שלה גדול יותר מהערך של החבילה עם הערך הקטן ביותר בתור.

**ילכו לאיבוד:** הסלאק של החבילה קטן שווה מכמות השורות בקובץ + הערך שלה נמוך יחסית לחבילות האחרות (כלומר לא תבחר בירידת שורות) ולכן כאשר נגיע ל-EOF היא תיעלם בגלל שכל פעם שאנחנו מקבלים חבילה בכל שורה אנחנו מורידים את הסלאק ב-1.

#### **שווה לו:**

אם הערך של הסלאק של החבילה שרוצה להכנס לתור הוא לפחות כמו הערך של הסלאק של החבילה שהאלגוריתם חושב להוציא והערך של החבילה שרוצה להכנס לתור הוא גם כן גדול שווה. או שהערך של הסלאק של החבילה גדול ביחס להפסד רווח, החבילה שנכניס אינה תלך לאיבוד (\***ילכו לאיבוד**).

הערה: כאן אפשר להכניס מודל של למידת מכונה כמו שיצויין בסוף הפסקה על מנת להחליט אם להחליף או לא.

#### **- מה נקבל בירידת שורה: $\infty$**

אם S לא ריקה: נסתכל על קבוצת כל הפקטות S אשר ילכו לאיבוד:  
נבחר את החבילה בעלת הערך הגדול ביותר מבין כל החבילות בעלות ערך הסלאק הקטן ביותר מכולן.  
אם S ריקה: אז נבחר את הפקטה בעלת הערך הגדול ביותר.

#### **- מה נקבל בסוף הקובץ: EOF**

נבחר את הפקטה בעלת הערך הגדול ביותר מכל הפקטות אשר בעלות ערך הסלאק הקטן ביותר.

#### **יתרונות:**

1. מקסום ערך קבלה (ה-value שנסכם).
2. לא מאבדים פקטות אלא אם כן יש כמה חבילות בעלות אותו ערך אשר מגיעות לסלאק 0, אבל הערך של החבילה עם הערך הגדול ביותר מבין כולן תתקבל.

#### **חסרונות:**

1. האלגוריתם יצטרך לקרוא קריאה מקדימה של הקובץ כדי לדעת איך לפעול.
2. (כמה ירידות שורה יש + הסתכלות על גודל התור + כמה חבילות נאבד בשלב ה-EOF)
3. הרבה תנאי קבלה או החלפה של חבילות.
3. מימוש.

בנוסף, בגלל שהאלגוריתם הוא אלגוריתם אשר "מקבל החלטות" בהינתן קלט חדש, אפשר בעצם ליצור מודל של מכונת למידה אשר זוכר מה היה עד עכשיו, וכל פעם לומד לפי איזה אלגוריתם ללכת, לפי אלגוריתם אשר בוחר את ערך הסאלק הקטן ביותר כמו באלגוריתם EDF או לחלופין את הערך הגבוה ביותר כמו אלגוריתם BD.

ולכן אם יהיה לנו המון מידע אז המכונה תוכל ללמוד באיזה אלגוריתם לבחור בהינתן כל חבילה וקובץ. (בהתחלה המכונה תוכל ללמוד ע"י סיווג של בן אדם או ע"י הערך שקבלנו בסוף, וכך היא תדע מה לעשות.)