

24 מרץ 2011

פרופ"ח ארוז פטרנק  
חגי ערן

## אלגוריתמים לניהול זיכרון - 236780 חורף תשע"א - מועד ב'

### הנחיות:

- הבחינה עם חומר סגור.
- אסור להחזיק פלאפונים ברשות הנבחנים בזמן הבחינה.
- כתבו בצורה מסודרת ונקייה ובכתב ברור.
- בבחינה 4 שאלות, משך הבחינה שלוש שעות.
- נמקו את כל תשובותיכם.
- כאשר מבקשים להציע פיתרון אלגוריתמי לאיזושהי בעיה, יש לנסות לתת פיתרון יעיל ככל היותר במקום ובזמן.

**בהצלחה !**

## שאלה 1: (30 נקודות) שאלה זו מתייחסת לאלגוריתם דחיסה של Jonkers.

- א. (3 נק) באלגוריתם של Jonkers יש שני מעברים על ה-heap. הסבירו מהו מנגנון השרשור. כלומר, הסבירו למה הוא משמש ותארו בדיוק מה עושים כשמגיעים למחונן שצריך לשרשר.
- ב. (3 נק) תארו מי יהיו הרכיבים בשרשרת שמתחילה ב-header של אובייקט כלשהו A לאחר סיום המעבר הראשון של האלגוריתם, ומה יהיה הסדר של הרכיבים בשרשרת.
- ג. (4 נק) תארו שני יתרונות של Jonkers על אלגוריתם דחיסה של Lisp2.
- ד. (4 נק) הסבירו מה הבעיה עם נסיון להזיז את האובייקטים בעת ההגעה אליהם במעבר הראשון.
- ה. (16 נק) בסעיף זה ננסה לשלב את מנגנון השרשור של nkersJo עם טבלת הבלוקים (ה-offset vector) של ה-Compressor. מטרתנו לייצר אלגוריתם דחיסה שעובר על ה-heap פעם אחת בלבד ובניגוד ל-Compressor, אינו זקוק למעבר מקדים על טבלה.
- לאלגוריתם מותר להשתמש בשירשור כמו Jonkers, ומותר לו להשתמש בטבלה חיצונית כמו ה-Compressor. כמו כן, האלגוריתם יכול להניח קיום של bitmap שבו יש ביט דלוק המסמן תחילה וסיום של כל אובייקט ב-heap. אבל, אסור לאלגוריתם לבצע יותר ממעבר יחיד על ה-heap ואסור לו להשתמש במבני נתונים נוספים. בסיום המעבר הזה, האובייקטים צריכים להיות דחוסים בתחילת ה-heap עם כל המצביעים מעודכנים.
- a. תכננו אלגוריתם לפי הדרישות הנ"ל. תארו את מהלך האלגוריתם החדש. תארו מה קורה במעבר על כל אובייקט ב-heap.
- b. האם הבעיה שתיארתם בסעיף ד' עולה גם כאן? אם כן – הסבירו כיצד ניתן לפתור אותה. אם לא – הסבירו מדוע הבעיה מסעיף ד' לא רלוונטית כאן.
- c. תארו יתרון אחד של האלגוריתם שהצעתם על הקומפרסור, וחסרון אחד שלו על הקומפרסור.

## שאלה 2: (21 נקודות) שאלה זו מתייחסת לאלגוריתם ה-Train.

- א. (4 נק) כזכור, זליגמן וגרארופ זיהו באג באלגוריתם הרכבת, שכאשר הוא קורה, האיסוף לא מסיים לעולם עם הרכבת הראשונה. תארו תוכנית המגדילה ככל האפשר את הסיכויים להופעת הבאג.
- ב. (8 נק) כזכור, באלגוריתם המקורי נמצא באג הנובע מכך שהמצביעים משתנים ע"י התוכנית בין איסוף קרון אחד למשנהו. נציע תיקון שונה מזה שהוצג ע"י Seligman-Grarup. עליכם לומר אם גם תיקון זה עובד כשורה (ולנמק) או להציע דוגמה נגדית המוכיחה שאלגוריתם זה לא מתקן את הבאג.
- התיקון: בעת ריצת התוכנית יש write barrier המעדכן את ה-remembered set. נסיף ל-write barrier בדיקה: אם השינוי של המצביע הנוכחי מבטל הצבעה אל אובייקט A הנמצא ברכבת הנוכחית שאנו אוספים, אז נסמן את האובייקט A כאובייקט מיוחד. בעת איסוף הקרון, נעתיק את כל האובייקטים המיוחדים אל הרכבת האחרונה ונוציא אותם מרשימת המיוחדים.
- בתשובתכם הפרידו בין שני מקרים: ה-write-barrier עובד גם על ה-roots, וה-write-barrier לא פועל על ה-roots.
- ג. (4 נק) נניח לשם סעיף זה שבכל קרון יש בדיוק 2 אובייקטים, בכל רכבת שני קרונות ויש שלוש רכבות. נניח שבשלב זה כל 12 האובייקטים אינם נגישים. תארו את "התצורה הטובה ביותר" של מצביעים בין האובייקטים, במובן ש-12 האובייקטים ייאספו במספר מינימלי של איסופים.
- ד. (5 נק) כיוון שהתוכנית משנה מצביעים בין איסוף קרון אחד למשנהו, תיתכנה תצורות שונות של מצביעים בין האובייקטים, לפי פעולת התוכנית. אבל נניח שמהרגע שבו האובייקטים הושמו בקרונות הנוכחיים שלהם התוכנית לא נגעה בהם. האם המבנה שהצעתם בסעיף הקודם עדיין אפשרי? אם לא, תארו מבנה שונה שיגרום למספר מינימלי של איסופים אבל אפשרי ללא התערבות של התוכנית (מהרגע שבו הועתקו האובייקטים לקרונותיהם הנוכחיים).

### שאלה 3: (24 נקודות) שאלה זו מתייחסת ל-reference counting.

- א. (4 נק) תארו את השיטה הבסיסית של איסוף זבל באמצעות reference counting.
- ב. (5 נק) תארו את השיטה שבה שומרים שדה קצר יותר עבור ה-reference count. מה קורה כשיש overflow? מה המשמעות עבור האיסוף? כיצד ניתן לתקן מוני מחוונים שעברו overflow?
- ג. (5 נק) בהרצאה למדנו את שיטת ה-software cache, ששומרת חלק מהמונים שעברו overflow בטבלה חיצונית. תארו את שיטת ה-software cache לטיפול במוני מחוונים שעברו overflow. מה צריך לשמור ב-cache כדי להיות מסוגלים לתקן מונה מחוונים שעבר overflow? האם יתכן מקרה שבו נצטרך לסגת לשיטה הבסיסית מסעיף ב? אם כן – תארו מקרה כזה, אם לא – נמקו.
- ד. (5 נק) תארו את השיטה המשתמשת בביט יחיד למונה המחוונים ושומרת אותו במחון. מה יתרונה של שיטה זו?
- ה. (5 נק) נניח שמנסים להשתמש ב-software cache בשילוב עם השיטה של סעיף ד'. האם מספיק לשמור ב-cache את אותו מידע שהצעתם בסעיף 3? או שצריך לשמור משהו נוסף? נמקו, ותארו את הדרך שבה מתקנים את מונה המחון לאובייקט שמספר המחוונים אליו ירד ל-1 ויכול לצאת מה-cache.

### שאלה 4: (25 נקודות) שאלה זו מתייחסת ל-Allocation Techniques.

בשאלה זו ננסה לבנות תוכנית שמכריחה את first-fit להשתמש בהרבה זיכרון. המגבלה של התוכנית היא שהשטח (מספר המילים) הכולל המוקצה בכל נקודה בזמן לא יכול לעבור את  $M$ .  $M$  הוא פרמטר ולשם נוחות נניח שהוא חזקה של 2. התוכנית תנסה לייצר פרגמנטציה, כדי שההקצאות שלה יתפסו כמה שיותר מקום. התוכנית תהיה בנויה משלבים.

- א. (2 נק) בשלב ראשון, (כשמתחילים מ-heap ריק) התוכנית תבקש להקצות  $M$  אובייקטים מגודל 1 (כלומר מילה אחת). כמה זיכרון תתפוס הקצאה זו? איך ייראה ה-heap לאחר ההקצאה?
- בשלבים הבאים התוכנית תקצה אובייקטים מגודל 2, 4, 8, וכיו"ב עד שתגיע לגודל המקסימלי אותו ניתן להקצות.
- ב. (3 נק) בשלב שני, התוכנית תנסה לשחרר כמה שיותר אובייקטים ב-heap כך שלא יהיה ניתן להקצות אובייקט מגודל 2 על השטח שהיה תפוס עד עתה. מה יהיו האובייקטים שישוחררו? מה השטח הכולל של האובייקטים ששוחררו?
- ג. (2 נק) לאחר השחרור, התוכנית תבקש מהאלוקטור להקצות כמה שיותר אובייקטים מגודל 2 (כזכור, אסור לה להקצות יותר מאשר  $M$  מילים סה"כ). כמה אובייקטים מגודל 2 תוכל התוכנית להקצות? איפה יקצה האלוקטור את האובייקטים האלו? תארו את מבנה ה-heap. ניתן לתאר בציור, אך צריך להיות הסבר מדויק מה כל סימון בציור אומר.
- ד. (8 נק) בשלבים הבאים התוכנית תשחרר אובייקטים שכבר הוקצו, ובכל שלב תקצה אובייקטים שגודלם כפול מגודל האובייקטים שהוקצו בשלב הקודם. הסבירו כיצד תפעל התוכנית בשחרור והקצאת אובייקטים בשלבים הבאים, תוך ניסיון להביא למקסימום את השטח שההקצאות תופסות ב-heap.
- ה. (5 נק) חשבו כמה מקום יתפסו ההקצאות ב-heap לאחר ההקצאה של האובייקט הגדול ביותר. הניחו שהגודל המקסימלי האפשרי להקצאה הוא  $n$  מילים ו- $n$  הוא חזקה של 2. התוצאה צריכה להיות תלויה ב- $M$  וב- $n$ . הסבירו את החישוב.
- ו. (5 נק) האם ניהול זיכרון הכולל compaction יכול להועיל למניעת התנפחות הצריכה של ה-heap ולאפשר שימוש ב- $M$  מילים ב-heap לסיפוק כל צרכי האלוקציה והשחרור של התוכנית שהצעתם? אם כן – מהו המספר המינימלי של הפעלות של compaction שדרוש על-מנת לאפשר זאת. אם לא – הסבירו מדוע תמיד יצטרכו יותר מ- $M$  מילים לסיפוק הדרישות של התוכנית.

ז. (2 נק) האם יש מדיניות הפעלת  $\text{compaction}$  שתאפשר לכל תוכנית (לא רק לזו שהצעתם) להסתפק בזיכרון של  $M$  מילים כאשר  $M$  הוא מספר המילים המקסימלי באובייקטים מוקצים בכל שלב של הריצה.