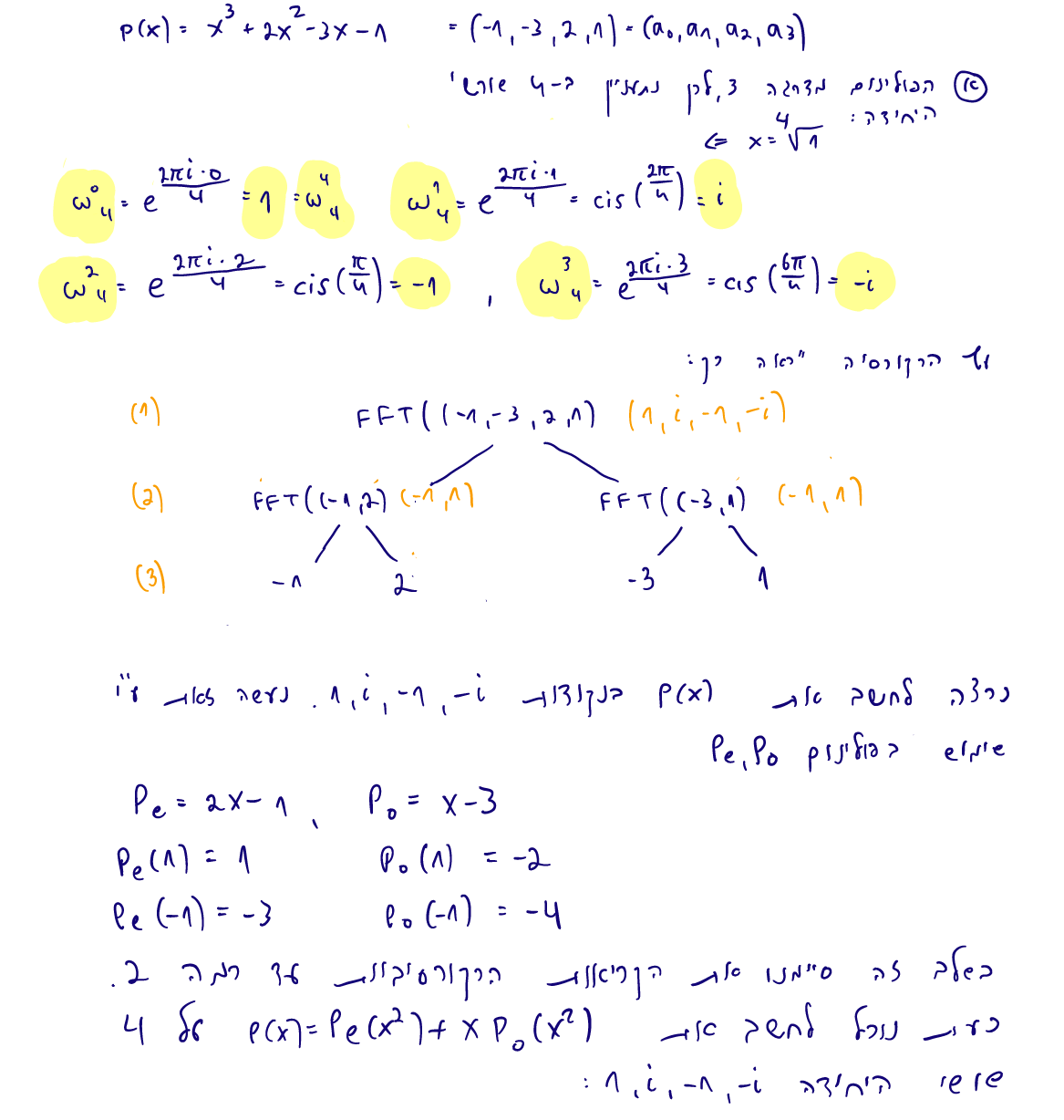
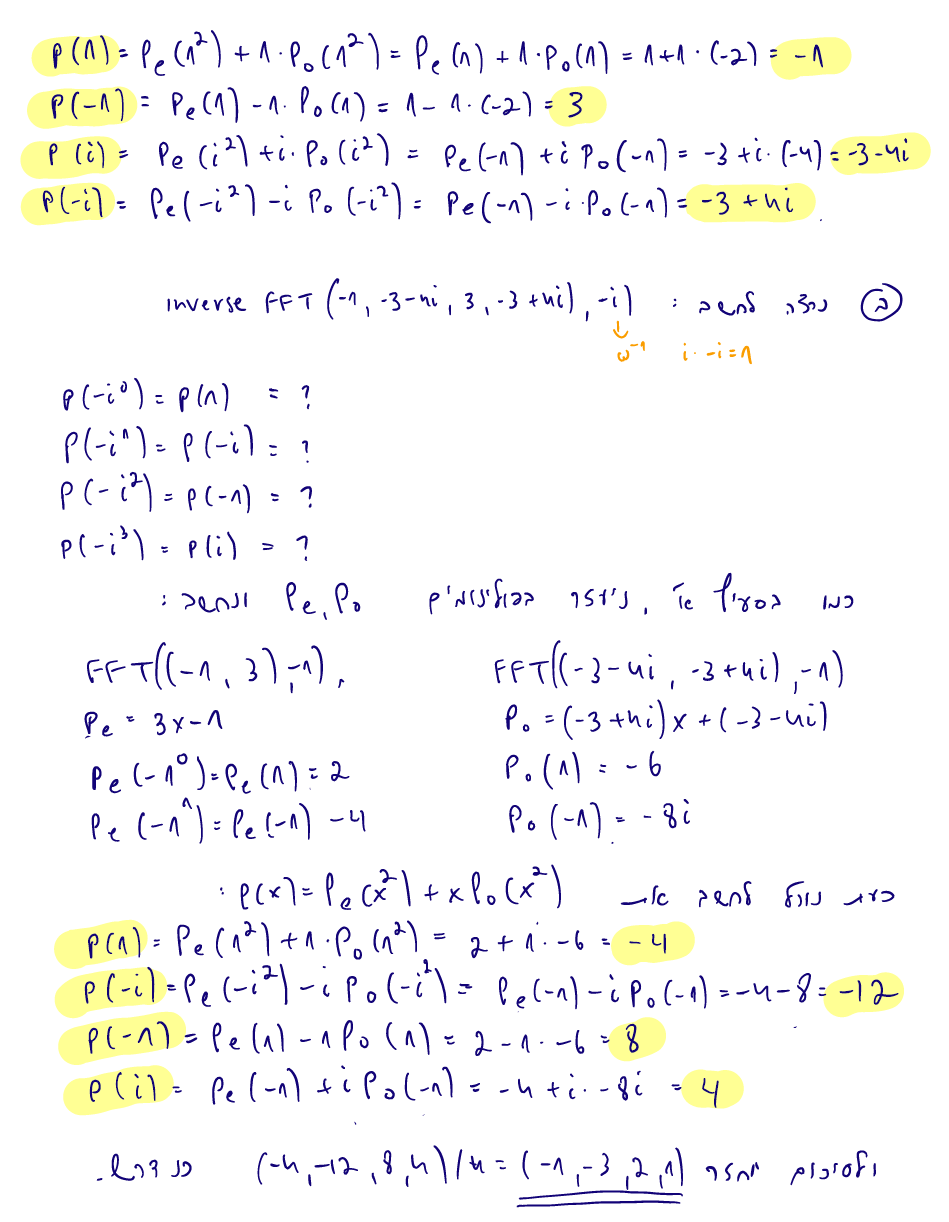
ממן 13

שאלה 1





שאלה 2

הרעיון הכללי של האלגוריתם –

יהיו שני מספרים x,y נתונים בייצוג בינארי בעל n סיביות. נחלק כל מספר ל בלוקים בגודל k (אם צריך נוסיף אפסים בהתאם). נסמן כל בלוק ב - . מכאן כי ניתן לייצג כל מספר בצורה הבאה : . לפנינו בעית כפל שני פולינומים מדרגה . נשתמש ב-FFT לכפל שני הפולינומים : ייצוג כל פולינום ב- שורשי היחידה, כפל בערכים אלו וחזרה לייצוג מקדמים בעזרת inverse FFT.

תיאור האלגוריתם –

1. בצע חלוקה של כל מספר ל- בלוקים בגודל k (אם צריך הוסף אפסים מובילים). אחסן את מקדמי כל בלוק במערך בצורה הבאה : ,
2. הרץ ,
3. בצע כפל שני הפולינומים מסעיף 2 ב-4 שורשי היחידה ואחסן את התוצאה במערך
4. הרץ והחזר תוצאה

הוכחת נכונות –

שלבי האלגוריתם כוללים חלוקת המספר הבינארי לבלוקים, הרצת FFT והכפלת שני פולינומים. ידוע כי ניתן לייצג מס' ע"י שימוש בחזקות הבסיס שלו, לכן בעצם החלוקה לא שינינו את המספר. בנוסף נתבסס על נכונות FFT מהספר וההערה בשאלה כי הכפלת הבלוקים לא מגדילה את אורכם.

ניתוח סיבוכיות –

1. חלוקת המספרים לבלוקים –
2. שתי הרצות FFT על המערכים בגודל + העבודה שמבוצעת בכל קריאה (לכל בלוקים ) -

לפי שיטת האב -

1. כפל שני הפולינומים המיוצגים בעזרת שורשי היחידה בשני בלוקים בגודל k -
2. כמו בסעיף 2

בסה"כ :

שאלה 3 – לא הצלחתי לפתור ☹

שאלה 4

מהנתונים בשאלה על r,s,t,u נוכל לראות כי לחישוב המטריצה c דרושות 8 פעולות כפל ועוד 4 פעולות חיבור מטריצות (כפל כל שורה בעמודה וסכום) ובכל קריאה רקורסיבית מחלקים את המטריצה המקורית ל-4 בלוקי מטריצות.

לכן נוכל לכתוב נוסחת נסיגה מתאימה :

לפי שיטת האב נקבל זמן ריצה .

אולם מהגדרת הכפל בעזרת המטריצות נוכל להשתמש ב-7 פעולות כפל במקום ב-8 (בנוסף הרבה יותר פעולות חיבור, אשר לא משנות את הזמן הריצה מכיוון שהן אורכות זמן קבוע) ולכן נוסחת הנסיגה החדשה תהיה :

לפי שיטת האב נקבל זמן ריצה