```
Re-Pair דחיסת דקדוק
                                                                       הגדרות לאלגוריתמים של עץ skeleton:
  כל עוד ישנו צמד חווים שחוזר על עצמו
                                      קוראים את הנוקחנו תו אחרי תו ומוחיפים לS
                                                                                                                                                                                                                             נכתב ע"י צבי מינץ ויואב ג ע"י צבי מינץ ויואב ג ני
בּגַּבָּי מרצה בפרופ' דנה שפירא ניג שפירא
                                                                                                                                                                                                                       נכתב ע"י צבי מינץ ויואב גרוס 🗓
                                                                                                                                                  Golomb encode(x,b)
                                  1. אם צמד תווים מופיע פעמיים בצד ימיו הוא יוחלף
                                                                        base(i) = 2(base(i-1) + n_{i-1}) Golomb decode(b)
 מצא את הזוג השכיח ביותר והחלף אותו
                                                              בכלל חדש.
                                                                               seq(i) = seq(i-1) + n_{i-1} \quad q \leftarrow Unary\_decode()-1;
                                                                                                                                                         a \leftarrow (x-1) \text{ div } b:
                               2. אם משתנה מופיע פעם אחת בצד ימין – נמחק אותו.
                  (בהינתן מילון עם מילות קוד באורך קבוע, יש להתאים מחרוזות) Tunstall
                                                                              diff(i) = base(i) - seq(i) r\leftarrowMinimal binary decode(b); r\leftarrowx-q·b;
                                                                                                                                                                                           מינמלי) E(C,P) ו-E(C,P) מינמלי)
                                                                                                                                                                                           E(C,P) = \sum_{i=1}^{n} p_i \cdot |c_i| אורך מילת הקוד הממוצעת:
                                                                                                                                                       I = Unary encode(q+1);
                                                                                                                return r+q·b;
                                                                          :Decoding Algorithm
                                             כל עוד נותר מקום לΣ תווים נוספים:
                                                                                                                                                       n = Minimal binary encode(r,b); return l·n
                                                                                                                                                                                                                       I(s_i) = -\log_2 p_i אינפורמציה:
                                                                         1. tree pointer<-root
         .1 עבור d מילת הקוד עם ההסתברות המקסי' במילון, הכנס את d\sigma: d מילון.
                                                                                                                                                                                             \overline{H(P) = -\sum_{i=1}^{n} p_i \log_2 p_i} אנטרופיה (חסם תחתון):
                                                                                                                              נבחר את b להיות Rice code: נבחר את
                                                                                                         Skeleton
                                             .2 אם Z \notin \Sigma מחק את d מהמילוו.
                                                                                                                                                                                                                            \forall C: H(P) < E(C,P) כלומר
                                                                                                                     חלק ראשון: קוד אונרי של (right shift k bits)) חלק ראשון: קוד אונרי של
                                                           קוד פיבונאצ'י 1
                                                                                                                                                                                            C' אזי קיים קוד K(C) = \sum_{i=1}^{n} 2^{-|c_i|} \le 1 אזי קיים קוד
                                                                         4. while i<length_of_string
                                                                                                                         חלק שני: k הביטים הנמוכים של הייצוג הבינרי של X-1
                   כתוב את המספר בבסיס פיבונאצ'י, הוסף 1 משמאל, הפוך את המספר.
                                                                                   if string[i]=0
                                                                                                                                                                                             יסר ש-C'-ו ו|C| = |C'|, E(C, P) = E(C', P) פרפיקסי
                                                                                                                                  (לא תמיד הבי יעיל) Shannon-Fano Algorithm:
 k+1 הן באורך F_{k+1} \leq j < F_{k+2} הקוד בטווח הקוד מידי. כל מידי. כל מידי. כל מילות הקוד בטווח
                                                                                        tree pointer<-left(tree pointer)
                                                                                                                                                                                             אזי לא קיים קוד פרפיקסי עם האורכים K(\mathcal{C}) > 1
                                                                                                                                                      מייו את ההסתברויות בסדר יורד.
                                                           2 קוד פיבונאצ'י 4.2.
                                                                                   else tree pointer<-right(tree pointer)
                                                                                                                                                                                                        מלא מלא יוצר עץ מלא K(\mathcal{C}) = 1. קוד שלם יוצר עץ מלא
                                                                                                                                        כל עוד קבוצת ההסתברות מכילה יותר מתו 1:
                                             Fib2(n+1) עבור, Fib2(1) = 1 4.3.
                                                                                   if value(tree_pointer)>0
                                                                                                                                                                                           (ניתן לפענוח בצורה יחידה) UD קוד חסר רישות \Rightarrow קוד
                                                                                                                         -נחלק את הקבוצה ל2 חלקים כך שסכום ההסתברויות בכל
                      . משמאל 10 כתוב את בבסיס פיבונאצ'י, הפוך את המספר, הוסף 10 משמאל n
                                                                                        codeword<-string [start ...
                                                                                                                          .0 קוד מיידי \Leftrightarrow קוד חסר רישות, קוד של \Leftrightarrow קוד חסר רישות חלק פחות או יותר זהה. -קבוצה אחת תקבל 1, והשנייה
                                                       (start+value(tree_pointer)-1)]
                                                                                                                                                                                                                        מבחו לזיהוי קוד יחודי: (אם"ם)
                                                                                        output<-table[I(codeword)-
                                                                                                                                               קוד הפמן: אלגוריתם לבניית עץ קנוני
       את לחפש אז לחפש את \sigma נמצא אז לחפש את – אפשר ע"י גרידי (אם \sigma נמצא אז לחפש את 4.3.2.
             וכו׳ ואז להתחיל מחדש),אפשר למצוא אופטימלי \sigma \cdot \sigma_2 diff[value(tree_pointer)]]
                                                                                                                          (l_1, \dots, l_n) אחרי הרצת הפמו היבלנו את כל האורכים

    Examine all pairs of codewords:

                                                                                                                                                                                                                                                                     \sigma היא רישא של \sigma x ולכן סיפא מתנדנדת היא \sigma אם \sigma היא מילת קוד מקורית – הקוד לא ייחודי
 4.3.3. ע"י רדוקציה לגרף, דחיסה אדפטיבית – המילון נבנה בזמן הקידוד
                                                                                        tree pointer←root
                                                                                                                          1.maxlenath ←האורר של המילה הארוכה ביותר
                                                                                                                                                                                        Construct a list of all codewords.
    ומכיל שלשות Sliding Window, Look Ahead Buffer עם – LZ77 4.3.4.
                                                                                        start<-start+value(tree pointer)
                                                                                                                         2. for i = 1 to maxlength do \{num[i] = 0\}
                                                                                                                                                                                     2. If there exist a codeword, a, which is a prefix of
                  (0,0,\sigma) כאשר לקדד תו בודד זה (off,len,symbol) 4.3.5.
                                                                                        i<-start
                                                                                                                                                                                         another codeword, b, add the dangling suffix to the
                                                                                                                          3. for i = 1 to n do \{num[l_i] + +\}
                                         : LZ77 אלגוריתם קידוד
                                                                         4.4.
                                                                                   else i++
                                                                                                                                                                                         list (if it is not there already), until:
                                                                                                                          4.firstcode[maxlenath] \leftarrow 0
         1. p\leftarrow 1 // The next character to be coded
                                                                                                                                                                                             You get a dangling suffix that is an original codeword \rightarrow the
                                                                                                    scaling אריתמטי בינארי עם 5. for <math>i=maxlength-1\ downto\ 1\ do
         2. while there is text remaining to be coded
                                                                                          y \leftarrow x; i \leftarrow 0
                                                                                                                          5.1.firstcode[i] \leftarrow (firstcode[i+1] +
                                                                                                                                                                                            There are no more unique dangling suffixes \rightarrow the code is
         2.1. search for the longest match for S[p..] in S[p-
                                                                                           while y > 0
                                                                                                                                                                num[i + 1])/2
         W...p-1] if the match occurs at position m with
                                                                                                                                                                                               מודלים: (ללמוד ולעשות הנחות על מבנה הטקסט)
                                                                                                 i \leftarrow i + 1;
         length I
                                                                                                                          6. for i = 1 to maxlength do
                                                                                                                                                                                                    Lossy - Lossless – יש 2 סוגים של דחיסות
ex.Video ex.Text
                                                                                                 if y < 1/2
         2.2 Output the triple (p-m, I,S[p+l])
                                                                                                                          6.1. nextcode[i] \leftarrow firstcode[i]
                                                                                                       b_i \leftarrow 0
                                                                                                                          7. for i = 1 to n do
                           length > offset הצבה עצמית – כאשר
                                                                                                                                                                                   (prelude אין |\Sigma| = 256(Ascii) כאשר p_i = \frac{1}{|\Sigma|} (אין
                           (AVL) שיפור, משתמש בעץ בינארי מאוזן – LZSS
                                                                                                       y \leftarrow 2y
                                                                                                                         7.1.codeword[i] \leftarrow nextcode[l_i]
                                                                                   0
                           ומחזיק ביט + (off,len) או ביט + אסקי (תו בודד)
                                                                                                 else if y > 1/2
                                                                                מוכל ב(1,
                                                                                                                          7.2.symbol[l_i, nextcode[l_i] - firstcode[l_i]] \leftarrow i
 w ← first char of input
                                                                                   \frac{1}{2}הטווח מוכל ב
                                                                                                                                                                                       |\Sigma|= כך p_i=rac{1}{|\Sigma|} בטקסט p_i=rac{1}{|\Sigma|}
                           Index | Phrase שימוש בעץ, trie שימוש בעץ – LZW
                                                                                                       b_i \leftarrow 1
 repeat
                      ומאותחל עם תווים בודדים, בנוסף מחזיק טבלה w | k
                                                                                                                          7.3.nextcode[l_i] + +
                                                                                                                                                                                   E(C,P)=H(P)+rac{8\cdot|\Sigma|+8}{|Text,Size|}ולכן ולכן אסקיE(C,P)=H(P)+rac{8\cdot|\Sigma|+8}{|Text,Size|}ולכן
                                                      אלגוריתם לקידוד:
                                                                                                       y \leftarrow 2y - 1;
 k ← next char
                                                                                                                                   אלגוריתם לפענוח של הפמן קנוני ע"י הטבלאות:
                                                                                              \leftarrow 0 for all j > i+1
   if (EOF) output code(w)
                                                                                                                                                                                    סמי סטטי עם הסתברויות עצמיות: p_i = rac{v_i}{m} כאשר מספר
                                                                                                                          1. v \leftarrow nextInputBit()// קבלת הביט הראשון
                                                                                                                         2.i = 1
                                                                                                                                                                                  רביר עולה עולה |p| בטקסט בגודל m. נציין ב-|s_i| כמה עולה לעביר
     if ((w \cdot k) \in Dictionary) then w \leftarrow w \cdot k
                                                                                                                         3. while v < firstcode[i] do
                                                                                                                                                                                                  E(C,P) = H(P) + \frac{8\cdot|\Sigma| + |\Sigma| \cdot |p| + 8}{|Text \, Size|}הסתברות ולכן
              output code(w)
                                                                                                pw = bound(s_i) \leftarrow \sum_{i=1}^{n-1} p_i
                                                                                                                         3.1. v \leftarrow 2v + nextInputBit()
              Dictionary ← w · k
                                                                          [0,1) קידוד - אריתמטי
                                                                                                                                                                                                       קודים: (רוצים למצוא קודים בסדר גודל לוגרתמי)
                                                                                                                         3.2. i + +
                                                                                                igh \_ bound (s_i) \leftarrow \sum_{i} 
                                                                                                                          אם יצאנו מהלולאה אזי v מכילה מילת קוד תקינה//
                                                                                                                                                                                                          קודים מסדר ראשון: ישנה תלות בין כל זוג תווים
                                                                        high \leftarrow 1.0
                                                                                                                          4.return\ symbol[i, v-firstcode[i]]
                                                                         while input symbols remain{
                                                                                                                                                                                                                       Unarv(X) = 1^{|X-1|} \cdot 0: Unary Code
אלגוריתם פענוח ל LZW: אתחל מילון עם תוים בודדים
                                                                            range ← high - low
                                                                                                                                          אלגוריתם לעדכוו עץ הפמו דינמי:
     OLD = first input code
                                                                                                                                                                                                 [\log_2 n] כל סימן מייצג מילת קוד באורך: Binary Code:
                                                                                                                        q = leaf(x_t)
     output translation of OLD
                                                                            high ← low + high bound(symbol) *range
                                                                                                                                                                                          יהיו 2^{\lceil \log_2 n \rceil} - n אם יש: Minimal Binary Code
                                                                                 ← low + low bound(symbol)*range
     while not end of input stream{
                                                                                                                             replace q by a parent 0-node with two 0-node
                                                                                                                                                                                           \lceil \log_2 n \rceil באורך באורך אורך ביטים ושאר באורך באורך אורך באורך באורך ו
       NEW = next input code
                                                                                                                             a = left child:
                                                                        Output any value in [low, high)
       if NEW is not in the string table
                                                                                                                        if q is a sibling of a 0-node
                                                                                                                                                                                       של מספר הביטים Unary און זה החלק הראשון החלק הראשון C_{v}: Elias
           S = translation of OLD
                                                                                                                             interchange q with the highest numbered leaf
                                                                                  אלגוריתם לפענוח קוד אריתמטי בטווח [0,1]:
                                                                                                                        of the same weight:
                                                                                                                                                                                             ללא '1' הראשון X ליצוג X והחלק השני זה קוד הבינארי
           S = S \cdot C
                                                                        encoded ← Get (encoded number)
                                                                                                                             increment q's weight by 1;
                                                                                                                             q = parent of q
                                                                                                                                                                                          X של מספר הביטים ליצוג \mathcal{C}_{\nu} און זה \mathcal{C}_{\kappa} של החלק הראשון זה \mathcal{C}_{\delta}
                                                                        do{ Find symbol whose range contains encoded
           S = translation of NEW
                                                                        Output the symbol
                                                                                                                             interchange q with the highest numbered node
                                                                                                                                                                                                       והחלק השני זה קוד הבינארי של X ללא '1' הראשוו
       output S
                                                                                                                        of the same weight;
                                                                        range ← high(symbol) – low(symbol)
       C = first character of S
                                                                                                                             increment q's weight by 1;
                                                                                                                                                                                                                                                              סדרי גודל:
                                                                        encoded \leftarrow (encoded – low(symbol))/range
       Translation(OLD) C to the string table
                                                                                                                             q = parent of q
                                                                                                                                                                                          C_{v}: 1 + 2 \cdot \left[\log_2 x \right] bits \right[C_{\delta}: 1 + 2\left[\log_2 \log_2 2x \right] + \left[\log_2 x \right] bits
       OLD = NEW }
                                                                                                                        increment q's weight by 1
```