<u>תכנון אלגוריתמים תרגיל 3 – דף תשובות</u>

תאריך הגשה: 6/12/19, בשעה 7:59 בבוקר. יש להגיש את העבודה במערכת ההגשה.

מתרגל אחראי: רון קורין

הוראות כלליות:

- כל עוד לא נאמר אחרת, כאשר הנכם מתבקשים לתאר אלגוריתם יש לספק:
  1. תיאור מילולי של האלגוריתם
  2. הוכחת נכונות
  3. ניתוח זמן-ריצה
- אלגוריתם עם זמן-ריצה אקספוננציאלי נחשב לא-יעיל ולכן בדרך כלל לא יתקבל.
- יש לכתוב את הפתרון רק בדף התשובות הנלווה לעבודה.
- **בכל שימוש במשפט שהוכח בכיתה, יש לצטט את המשפט באופן מדויק.**

אנא הגישו רק חלק זה. <u>אל תחרגו מהמקום המוקצה לתשובה!</u>

שימו לב! ההגשה היא לSubmission System

שאלה 1

סעיף א'

1.	$Sol(i,j,k') = \{(<x_0,y_0>,\dots,<x_t,y_t>) \mid (<x_0,y_0>=<i,j>),(t=k')OR(y_t=n),$
2.	$\forall p\ , 0 \le p \le t-1 : \lvert x_{p+1}-x_p \rvert + \lvert y_{p+1}-y_p \rvert \le 1\ , 1 \le y_{p+1}-y_p,$
3.	$\forall m\ , 0 \le m \le t : 1 \le x_m, y_m \le n\}$
4.	$COST(Sol(i,j,k')) = \sum_{v=0}^{t} B[x_v][y_v]$
5.	$OPT(i,j,k') = \max\{COST(Sol(i,j,k'))\}$
6.	
7.	
8.	
9.	

סעיף ב'

1.	$OPT(i,n,k') = B[i][n]\ ,\ OPT(i,j,0) = B[i][j]$
2.	$OPT(i,j,k') = B[i][j] + OPT(i',j',k'-1) \mid \lvert i'-i \rvert + \lvert j'-j \rvert \le 1\ , 1 \le i' \le n\ , j < j' \le n$
3.	
4.	
5.	
6.	

סעיף ג'

j=n או k'=0 , אזי המשחק נגמר ולכן $Sol(i,j,k') = (<i,j>)$ ומכאן ש - $OPT(i,j,k') = B[i][j]$

<u>*טענה:</u> קבוצת הפתרונות $Sol(i,j,k')$ הינה מהצורה $(<i,j>, Sol(i',j',k'-1))$ כך ש -

$1 \le i' \le n , j < j' \le n$, $|i'-i| + |j'-j| \le l$, זאת ישירות מהגדרתה, לפיה האיבר הראשון בסדרה הינו $<i,j>$

והאיבר השני מוגדר לפי האיבר הראשון ובתחום מסוים ע"פ אי השוויונות לעיל, ומספר הצעדים שנותרו ירד ב – 1.

$OPT(i,j,k') = \max\{COST(Sol(i,j,k'))\} =$ (ע"פ הגדרת OPT)

$\max\{COST(<x_0,y_0>,\dots,<x_t,y_t>) \mid defined\ by\ clause\ a\} =$ (ע"פ הגדרת SOL)

$\max\{COST(<i,j>,\dots,<x_t,y_t>) \mid defined\ by\ clause\ a\} =$

$\max\{B[i][j] + COST(<x_1,y_1>,\dots,<x_t,y_t>) \mid defined\ by\ clause\ a\} =$

$B[i][j] + \max\{COST(<x_1,y_1>,\dots,<x_t,y_t>) \mid defined\ by\ clause\ a\} =$

$B[i][j] + \max\{COST(Sol(i',j',k'-1))\} =$ (ע"פ טענה *)

$B[i][j] + OPT(i',j',k'-1) \mid 1 \le i' \le n , j < j' \le n , |i'-i| + |j'-j| \le l$

.39
.40
.41
.42
.43
.44
.45
.46
.47
.48
.49
.50
.51
.52
.53
.54
.55
.56
.57
.58
.59
.60

סעיף ד'

1. <u>תיאור אלגוריתם:</u>
2. נגדיר: מערך תלת ממדי $M[n][n][k+1]$.
3. אתחול: $M[i][j][0] = B[i][j]$ לכל $1 \leq i,j \leq n$, $M[i][n][k'] = B[i][n]$, לכל $1 \leq i \leq n$, $0 \leq k' \leq k$.
4. צעד:
$for(j = n-1;\ 1 \leq j;\ j--)\{$
$for(i = 1;\ i \leq n;\ i++)\{$
$for(k' = k; 0 < k'; k'--)\{$
$M[i][j][k'] = B[i][j] + \max\{M[i'][j'][k'-1] \mid \|i'-i\| + \|j'-j\| \leq l\ , 1 \leq i' \leq n\ , j < j' \leq n\}\}\}\}$
5. החזר: $M[1][1][k]$
6.
7. <u>ניתוח זמן ריצה:</u> אתחול האלגוריתם מתבצע בזמן של $O(n^2 + k*n)$. בצעד האלגוריתם יש n איטרציות, של n
8. איטרציות, של k איטרציות שבכל אחת אנו מבצעים לכל היותר $1+3+5+7+\cdots+(2l-1) = 2l^2$, לכן זמן
9. הריצה של הצעד הינו $O(l^2 * k * n^2)$.
10. בסה"כ קיבלנו כי זמן הריצה של האלגוריתם כולו הינו $O(l^2 * k * n^2)$.

סעיף ה'

<u>תיאור האלגוריתם:</u>
נגדיר <>S כסדרת הפתרון ו - $temp =< 1,1 >, step = k$.
S=<temp>.
כל עוד $step > 0$ וגם הקורדינטה השנייה של temp קטנה מ - n:
$temp =< i,j >$ כך ש - $M[i][j][step-1]$ מקסימלי מבין כל הצעדים החוקיים.
S=<S,temp>
step--
החזר את S.
<u>ניתוח זמן ריצה:</u> הלולאה מתרחשת ב - $O(n)$, ובכל פעם מבצעים בדיקה בזמן לכל היותר $2l^2$, ולכן זמן ריצת האלגוריתם הינו $O(l^2 * n)$.

שאלה 2

סעיף א'

נסמן: $\mathrm{Cat}_{\mathrm{Zion}} =< z_1, \dots, z_n >$ ונגדיר: Super_{Sub} = $\mathbf{Sol(j)}$ עבור זוג המחרוזות: $\mathrm{Cat}_{\mathrm{Lucky}}$, $< z_1, \dots, z_j >$.
$(1 \le j \le n)$.
בנוסף, נגדיר את $\boldsymbol{OPT(j)}$ = המשקל המקסימלי של Sol(j) לפי פונקציית המשקל w $(1 \le j \le n)$.

סעיף ב'

.1	$OPT(1) = w(z_1)$ במידה ו - z_1 קיים ב - Cat_{Lucky}. אחרת, $OPT(1) = 0$.
.2	$OPT(j) = \max\{OPT(j-1), f(j)\}$ $(1 \le j \le n)$ כך ש - $f: \{1,..,j\} \to \mathbb{R}^+$ המוגדרת באופן הבא:
.3	$f(j) =$ משקל ה - $Super_{Sub}$ המקסימלי עבור זוג המחרוזות: Cat_{Lucky} , $< z_1, \dots, z_j >$ שמסתיימים ב - z_j (תת מחרוזות
.4	של Cat_{Zion} מהצורה $< \cdots, z_{j-1}, z_j >$ שהינה תת סדרה של Cat_{Lucky}). נסמן את קבוצת ה - $Super_{Sub}$'ים הללו $A(j)$.
.5	לכן $f(j) = \max\{W(A(j))\}$
.6	
.7	
.8	

סעיף ג'

.1	$<>$ או $Sol(1) =< z_1 >$ לכן 0 או $OPT(1) = w(z_1)$ (z_1 לא קיים ב - Cat_{Lucky} **או** z_1 קיים ב - Cat_{Lucky}).
.2	*טענה: $Sol(j) = A(j) \cup Sol(j-1)$. זאת משום שניתן לחלק את $Sol(j)$ לשתי קבוצות, האחת מכילה את z_j והשנייה לא.
.3	תת הקבוצה שלא מכילה את z_j הינה בדיוק $Sol(j-1)$. הקבוצה השנייה שכן, הינה קבוצה של תת מחרוזות מהצורה
.4	$< \cdots, z_{j-1}, z_j >$ שהינן תת סדרות של Cat_{Lucky}, בדיוק לפי הגדרת $A(j)$.
.5	$OPT(j) = \max\{W(Sol(j)\} =$ \\\\מהגדרת $OPT(j)$
.6	$= \max\{W(Sol(j-1) \cup A(j))\} =$ \\\\לפי *
.7	$= \max\{\max\{W(Sol(j-1))\}, \max\{W(A(j))\}\} =$ \\\\קבוצות זרות
.8	$= \max\{OPT(j-1), f(j)\}$ \\\\מהגדרת $f(j)$, $OPT(j)$
.9	
.10	
.11	
.12	
.13	
.14	
.15	
.16	
.17	
.18	
.19	
.20	
.21	
.22	
.23	
.24	
.25	
.26	

.27
.28
.29
.30
.31
.32
.33
.34
.35
.36
.37
.38
.39
.40
.41
.42
.43
.44
.45
.46
.47
.48
.49
.50
.51
.52
.53
.54
.55
.56
.57
.58
.59
.60
.61
.62
.63
.64
.65

66.
67.
68.
69.
70.
71.
72.
73.
74.
75.

סעיף ד'

1. נגדיר $M[n]$.
2. אתחול: נחפש את z_1 ב - Cat_{Lucky}. במידה ונמצא נאתחל $M[1] = w(z_1)$. אחרת, $M[1] = 0$.
3. צעד: לכל j בין 2 ל – n :
4. T=0.
5. נחפש מהתו האחרון של Cat_{Lucky} את z_j. בפעם הראשונה שמצאנו (אם בכלל) $T = T + w(z_j)$, ונמשיך מאותה נקודה
6. בחיפוש אחר z_{j-1}. בפעם הראשונה שמצאנו (אם בכלל) $T = T + w(z_{j-1})$, ונמשיך מאותה נקודה בחיפוש אחר z_{j-2}.
7. וכן הלאה, עד שלא מצאנו ב - Cat_{Lucky} תו כלשהו שחיפשנו **או** שחיפשנו ומצאנו את z_1.
8. $M[j] = \max\{M[j-1], T\}$.
9. החזר: $M[n]$.
10. נסמן את גודל המחרוזת Cat_{Lucky} ב – m. אתחול האלגוריתם מתבצע בזמן של $O(m)$. בצעד האלגוריתם אנו מבצעים
11. n איטרציות שכל אחת מהן מתבצעת ב - $O(m)$ (מכיוון שאנו רצים פעם אחת לכל היותר על תווי המחרוזת Cat_{Lucky}),
12. ובסה"כ קיבלנו כי זמן הריצה של האלגוריתם הוא $O(m * n)$.
13.
14.
15.
16.

סעיף ה'

1. נחפש מסוף מערך M את הפעם הראשונה שהערך במערך משתנה (נסמן כי השינוי הראשון קרה בפעם הראשונה בין התא
2. ה – t לתא ה – t-1).
3. נחפש מהתו האחרון של Cat_{Lucky} את z_t. בפעם הראשונה שמצאנו (אם בכלל) נשרשר את z_t לתחילת מערך הפלט שלנו
4. O, ונמשיך מאותה נקודה בחיפוש אחר z_{t-1}. בפעם הראשונה שמצאנו (אם בכלל) נשרשר את z_{t-1} לתחילת O, ונמשיך

17. מאותה הנקודה בחיפוש אחר z_{t-2}. וכן הלאה, עד שלא מצאנו ב - $\mathrm{Cat}_{\mathrm{Lucky}}$ תו כלשהו שחיפשנו או שחיפשנו ומצאנו את
5. z_1. נחזיר את O.
6. האלגוריתם עובר פעם אחת לכל היותר על מערך בגודל n ולאחר מכן פעם אחת לכל היותר על מחרוזת $\mathrm{Cat}_{\mathrm{Lucky}}$, ולכן
7. זמן הריצה שלו הינו $O(m+n)$.
8.
9.
10.
11.
12.
13.
14.
15.
16.
17.
18.
19.
20.
21.
22.
23.
24.
25.
26.
27.
28.
29.
30.

שאלה 3

סעיף א'

1. **<u>תיאור האלגוריתם:</u>**
2. לכל $\mathrm{e} \in \mathrm{E}$ נגדיר $\mathrm{w}': \mathrm{E} \rightarrow \mathbb{R}^{+} \cup \{0\}$ באופן הבא: $w'(e) = w(e) + 1$.
3. נפעיל את אלגוריתם Dikjstra עבור המופע $G = (V, E)$, $\mathrm{w}': \mathrm{E} \rightarrow \mathbb{R}^{+} \cup \{0\}$ ו - $\mathrm{s} \in \mathrm{V}$.
4. נחזיר את משקל המסלול הקל ביותר מ - s שקיבלנו מאלגוריתם Dikjstra לכל $v \in \mathrm{V}$.
5.
6. **<u>הוכחת נכונות:</u>**
7. <u>טענה מרכזית:</u> האלגוריתם מחזיר לכל $\mathrm{v} \in \mathrm{V}$ את ה"משקל אורך" המינימלי של מסלול מ – s ל – v.
8. <u>טענת עזר:</u> ה"משקל אורך" של מסלול p בגרף G = סכום המשקלים של קשתות p לפי w'.
9.

הוכחת טענה מרכזית: יהי $v \in V$ ונסמן את המסלול מ – s ל – v בעל ה"משקל אורך" שהתקבל מהאלגוריתם ב – p. נניח בשלילה שקיים מסלול אחר p' מ – s ל – v כך שה"משקל אורך" שלו קטן ממשקל האורך של p. ע"פ טענת העזר, סכום המשקלים של קשתות p' קטן ממש מסכום המשקלים של קשתות p לפי w', בסתירה לנכונות Dijkstra.

הוכחת טענת עזר: יהי מסלול $p = v_1, \dots, v_{k+1}$ בגרף G. ע"פ ההגדרה ה"משקל אורך" של p הינו משקלו לפי w + אורכו, כלומר: $\sum_{i=1}^{k} w((v_i, v_{i+1})) + k$. משמע, קיבלנו כי ה"משקל אורך" של p= $\sum_{i=1}^{k} w((v_i, v_{i+1})) + k = \left[w\big((v_1, v_2)\big) + 1\right] + \dots + \left[w\big((v_k, v_{k+1})\big) + 1\right] = \sum_{i=1}^{k} w'\big((v_i, v_{i+1})\big) =$ סכום המשקלים של קשתות p לפי w'.

ניתוח זמן ריצה: בניית $w': E \to \mathbb{R}^+ \cup \{0\}$ מתבצעת בזמן $O(|E|)$ והפעלת אלגוריתם Dijkstra מתבצע בזמן $O(|E| + |V| \log|V|)$ ובסה"כ האלגוריתם כולו מתבצע בזמן של $O(|E| + |V| \log|V|)$.

סעיף ב'

תיאור האלגוריתם:

ממיר הקלט: ניצור גרף $G' = (V', E')$ באופן הבא:
עבור כל $v_i \in V$ ($1 \le i \le |V|$) ניצור לכל $1 \le j \le |V|$ קודקוד $v_i^j \in V'$.
בנוסף, עבור כל $(v_p, v_t) \in E$ לכל $1 \le j < |V|$ ניצור שתי קשתות $(v_p^j, v_t^{|V|}), (v_p^j, v_t^{j+1}) \in E'$ - נשים לב כי הקשת $(v_p^{|V|-1}, v_t^{|V|})$ נוצרת פעם אחת ולא פעמיים!
לבסוף, נגדיר $w': E' \to \mathbb{R}^+ \cup \{0\}$ באופן הבא: $w'((v_p^j, v_t^{|V|})) = w'((v_p^j, v_t^{j+1})) = w((v_p, v_t)) + [j^2 - (j-1)^2]$.

קופסא שחורה: נפעיל את אלגוריתם Dikjstra עבור המופע $G' = (V', E')$, $w': E' \to \mathbb{R}^+ \cup \{0\}$ ו - $v_p^1 \in V'$ (כך ש - $s = v_p \in V$).

ממיר הפלט:
לכל $v_i \in V$ נחזיר את משקל המסלול הקל ביותר מ - v_p^1 שקיבלנו מאלגוריתם Dikjstra ל - $v_i^{|V|}$.

הוכחת נכונות:

טענה מרכזית: האלגוריתם מחזיר לכל $v \in V$ את ה"משקל אורך בריבוע" המינימלי של מסלול מ – s ל – v.

טענת עזר: יהי $p = v_1, \dots, v_t$ מסלול פשוט בגרף G, $1 < t \le |V|$, אז ה"משקל אורך בריבוע" של p = סכום המשקלים של $v_1^1, v_2^2, \dots, v_{t-1}^{t-1}, v_t^{|V|}$ לפי w'.

הוכחת טענה מרכזית: יהי $v = v_i \in V$ ונניח בשלילה שקיים מסלול מ – s ל – v בעל "משקל אורך בריבוע" קטן ממש מזה שהחזיר האלגוריתם נסמנו $p = s = v_p, \dots, v = v_i$. אז ע"פ טענת העזר סכום המשקלים של $p' = v_p^1, \dots, v_i^{|V|}$ לפי w' קטן ממש מהמשקל שהחזיר האלגוריתם בסתירה לנכונות Dikjstra.

הוכחת טענת עזר: יהי $p = v_1, \dots, v_t$ מסלול פשוט בגרף G, $1 < t \le |V|$, אז ה"משקל אורך בריבוע" של p =

$$\sum_{i=1}^{t-1} w((v_i, v_{i+1})) + (t-1)^2 = \sum_{i=1}^{t-1} w((v_i, v_{i+1})) + 1^2 - 0^2 + 2^2 - 1^2 + 3^2 - 2^2 + \dots + (t-1)^2 - (t-2)^2 =$$

$$= w((v_1, v_2)) + (1^2 - 0^2) + \dots + w((v_{t-1}, v_{t-2})) + ((t-1)^2 + (t-2)^2) =$$

$$w'((v_1^1, v_2^2)) + \dots + w'((v_{t-2}^{t-2}, v_{t-1}^{t-1})) + w'((v_{t-1}^{t-1}, v_t^{|V|}))$$

25. <u>ניתוח זמן ריצה:</u> שלב תרגום הקלט כולל שכפול כל קודקוד וצלע בגרף|V| פעמים ומכאן שעלותו $O(\|V\| * (\|V\| + \|E\|))$.
26. אלגוריתם Dijktra מתבצע בזמן של $O(\|V\| * \|E\| + \|V\|^2 * log(\|V\|^2))$.
27. לאחר מכן שלב תרגום הפלט מתבצע ב $O(1)$.
28. בסה"כ קיבלנו שזמן הריצה של האלגוריתם הינו $O\left(\|V\| * \|E\| + \|V\|^2 * log(\|V\|^2)\right) = O(\|V\| * \|E\| + \|V\|^2 * log(\|V\|)$.
29.
30.
31.
32.
33.
34.
35.
36.
37.
38.
39.
40.
41.
42.
43.
44.
45.
46.
47.
48.
49.
50.
51.
52.
53.
54.
55.
56.
57.
58.
59.
60.
61.
62.
63.

.64	
.65	

סעיף ג'

.1	**<u>תיאור האלגוריתם:</u>**
.2	<u>ממיר הקלט:</u> ניצור גרף $G' = (V', E')$ באופן הבא:
.3	עבור כל $v_i \in V$ ($1 \le i \le \|V\|$) ניצור לכל $1 \le j \le 11$ קודקוד $v_i^j \in V'$.
.4	עבור כל $(v_p, v_t) \in E$: אם $w((v_p, v_t)) = x$ אז לכל $1 \le j \le 10$ ניצור $(v_p^j, v_t^{j+1}) \in E'$, אחרת, אם $w((v_p, v_t)) > 0$
.5	אז לכל $1 \le j \le 11$ ניצור $(v_p^j, v_t^j) \in E'$.
.6	לבסוף, נגדיר $w': E' \to \mathbb{R}^+ \cup \{0\}$ לפי: לכל $(v_p^i, v_t^j) \in E'$, אם $i = j$ אז $w'((v_p^i, v_t^j)) = w((v_p, v_t))$, אחרת
.7	$w'((v_p^i, v_t^j)) = 0$.
.8	<u>קופסא שחורה:</u> נפעיל את אלגוריתם Dikjstra עבור המופע $G' = (V', E')$, $w': E' \to \mathbb{R}^+ \cup \{0\}$ ו - $v_p^1 \in V'$ (כך ש -
.9	$s = v_p \in V$).
.10	<u>ממיר הפלט:</u> לכל v_t^j נסמן ב - $d(v_p^j)$ את המשקל שהחזיר עבורו האלגוריתם. לכל $1 \le j \le 11$ נסמן:
.11	$d'^{(v_p^j)} = d(v_p^j) + (j-1) * x$. לכל $v_t \in V$ נחזיר את המינימום מבין הקבוצה $C = \{d'(v_t^1), \ldots, d'(v_t^{11})\}$.
.12	
.13	**<u>הוכחת נכונות:</u>**
.66	<u>טענה ראשית:</u> האלגוריתם מחזיר לכל $v \in V$ את ה"משקל עד 10 שליליות" המינימלי של מסלול מ – s ל – v.
.14	<u>טענת עזר:</u> אם $p = s, \cdots, v_t$ מסלול בעל "משקל h שליליות" מינימלי מ – s ל – v אז
.15	$p' = s^1, \ldots, v_t^{h+1}$ הוא מסלול בעל משקל מינימלי מ - s^1 ל - v_t^{h+1} (כל קשת שלילית "מקדמת" אותנו קבוצת קודקודים).
.16	ומשקל p שווה $d'(v^{h+1})$
.17	
.18	<u>הוכחת טענה ראשית:</u> יהי $v \in V$ ונניח בשלילה כי קיים מסלול מ – s ל – v בעל "משקל עד 10 שליליות", y', בעל h'
.19	קשתות שליליות ומשקל מינימלי, הקטן ממש מזה שהחזיר האלגוריתם, y. מסלול זה הינו מינימלי לפי h' בפרט ולכן
.20	ע"פ טענת העזר, קיים מסלול $p' = s^1, \ldots, v^{h'+1}$
.21	בעל משקל מינימלי מ - s^1 ל - $v^{h'+1}$. נשים לב כי משקל מסלול זה הינו $d(v^{h'+1})$, ולפי הגדרת האלגוריתם
.22	$d'^{(v^{h'+1})} \ge d'(v^{h+1})$. אולם ע"פ טענת העזר $y' = d'^{(v^{h'+1})} \ge d'^{(v^{h+1})} = y$ בסתירה להנחה.
.23	<u>הוכחת טענת עזר:</u>
.24	יהי $p = s, \cdots, v_t$ מסלול בעל "משקל h שליליות" מינימלי מ – s ל – v, נשים לב שכל מסלול מ-s^1 ל - v_t^{h+1}
.25	מייצג מסלול בעל h קשתות שליליות מ –s ל-v ב- G, אז אם p הינו מסלול בעל משקל מינימאלי בעל h קשתות שליליות
.26	אז $p' = s^1, \ldots, v_t^{h+1}$ הוא מסלול בעל משקל מינימלי מ - s^1 ל - v_t^{h+1} (כל קשת שלילית "מקדמת" אותנו קבוצת קודקודים).
.27	בנוסף עבור כל קשת ב p, סכמנו את משקלה במידה והינה חיובית, אחרת סכמנו בסוף האלגוריתם. ולכן משקל p
.28	שווה $d'^{(v^{h+1})}$.
.29	
.30	

31. **<u>ניתוח זמן ריצה:</u>** שכפול הגרף 11 פעמים הינו $O(\lvert E\rvert + \lvert V\rvert)$. הפעלת דייסקטרה מתבצעת ב:$O(\lvert E\rvert + \lvert V\rvert \log(\lvert V\rvert))$
32. מציאת האיבר המינימאלי מתבצעת ב $O(\lvert V\rvert)$. ומכאן שסך הכל זמן הריצה הכולל הינו$O(\lvert E\rvert + \lvert V\rvert \log(\lvert V\rvert))$.
33.
34.
35.
36.
37.
38.
39.
40.
41.
42.
43.
44.
45.
46.
47.
48.
49.
50.
51.
52.
53.

שאלה 4

1. נשלח לבעיית הארנק את n הפריטים שקיבלנו בבעיית התרמיל, ו - $P = \sum_{i=1}^{n} p_i$. נסתכל על טבלת התכנון הדינמי
2. $M[n+1][p+1]$ שקיבלנו במהלך ריצת האלגוריתם של בעיית הארנק, ונחפש מהעמודה ה - $p + 1$ עד לעמודה ה - 0
3. את הפעם הראשונה (נניח שנרוץ מהשורה ה – 0 לשורה ה – n+1) שהערך בטבלה קטן או שווה ל – W. נסמן את התא
4. שקיבלנו ב - $M[j][T]$. נפעיל את האלגוריתם המופיע בשלב השחזור של בעיית הארנק עם המופע: $Reconstruct(j, T)$.
5. **<u>הוכחת נכונות:</u>**
6. <u>טענה ראשית:</u> האלגוריתם מחזיר פתרון חוקי במחיר מקסימלי
7. <u>הוכחת נכונות:</u> מנכונות אלגוריתם בעיית הארנק, וע"פ הגדרת החיפוש בטבלת התכנון הדינמי שהגדרנו באלגוריתם שלנו,
8. נבחין כי מצאנו את המחיר המקסימלי שניתן להגיע אליו בעזרת פריטי הקלט מבלי לעבור את W הנתון. בנוסף, אלגוריתם
9. השחזור משחזר עבורנו את הפתרון שבעזרתו מגיעים למשקל המקסימלי מבלי לעבור את W. נבחין כי תמיד קיים פתרון
10. כזה, מכיוון שלא לקחת פריטים כלל זהו פתרון חוקי ו - $M[0][0] = 0$.
11.
12. **<u>ניתוח זמן ריצה:</u>**
13. תרגום הקלט מתבצע בזמן של $O(n)$. בניית טבלת התכנון הדינמי מתבצעת בזמן של $O(n * \sum_{i=1}^{n} p_i)$ ע"פ אלגוריתם
14. לפתרון בעיית הארנק. חיפוש אחר התא הראשון בטבלה שמשקלו קטן שווה מ – W מתבצע בזמן של $O(n * \sum_{i=1}^{n} p_i)$.

15. שחזור הפתרון האופטימלי מתבצע בזמן של $O(n)$ ע"פ אלגוריתם לפתרון בעיית הארנק.
16. בסה"כ קיבלנו כי זמן הריצה של האלגוריתם הינו $O(n * \sum_{i=1}^{n} p_i)$.
17.
18.
19.
20.
21.
22.
23.
24.
25.
26.
27.
28.
29.
30.
31.
32.
33.
34.
35.
36.
37.
38.
39.
40.
41.
42.
43.
44.
45.