תכנון אלגוריתמים תרגיל 3 – דף תשובות

שם:	מאור יעקב אסייג	ת.ז:	318550746
שם:	רפאל חי שטרית	ת.ז:	204654891

אנא הגישו רק חלק זה. אל תחרגו מהמקום המוקצה לתשובה!

שימו לב! ההגשה היא לSubmission System

שאלה 1

סעיף א'

1. אבחנה:
2. עבור $k > 0$ לכל פתרון ב$Sol(k)$ קיים $i \in \{1, \dots, k\}$ כך שהקפיצה האחרונה בפתרון
3. היא מנקודה $k - i$ ל-k
4.
5. טענה:
6. עבור$k > 0$, $i \in \{1, \dots, k\}$ קבוצת הפתרונות ב-$Sol(k)$ בהם הצפרדע קופצת מהנקודה
7. $k - i$ ישירות לנקודה k הינה בדיוק קבוצת כל הפתרונות מהצורה $S_o \cup \{i\}$ עבור
8. $S_o \in Sol(k-i)$
9.
10. הוכחת הטענה שלעיל:
11. נוכיח שוויון בין 2 הקבוצות הבאות :
12. $\mathrm{A} = \{\mathrm{I}: \mathrm{I} = \mathrm{S}_0 \cup \{i\} \| S_0 = (S_1, .., S_m),\ i + \sum_{j=1}^{m} S_j = k,\ S_0 \in Sol(k-i)$
13. $\mathrm{B} = \{\mathrm{I}: \mathrm{I} \subseteq Sol(k)\}$
14.
15. <= $\mathrm{A} \subseteq \mathrm{B}$: יהי $I \in A$, אזי $\mathrm{I} = \mathrm{S}_0 \cup \{i\}$. $S_0 \in Sol(k-i)$
16. ולכן $\mathrm{S}_0 \subseteq \{1,2, .., k-i\}$, כלומר $\mathrm{I} = (S_1, .., S_m, i) \| i + \sum_{j=1}^{m} S_j = k$
17. ולכן $\mathrm{I} \in Sol(k)$ ומכאן $\mathrm{A} \subseteq \mathrm{B}$.
18.

=> B ⊆ A : יהי $I \in B$, אזי $I \in Sol(k)$ ולכן $I \subseteq \{1,2,..,k\}$,

לפי האבחנה לכל I קיים i כך ש $I = S_0 \cup \{i\}$ כאשר i מהווה את הקפיצה

האחרונה מ $k-i$ ל k.

אזי I גם מקיים $I \in A$, ומכאן $B \subseteq A$.

=> $B \subseteq A, A \subseteq B \rightarrow A = B$ ■

הוכחת נכונות הנוסחא על סמך הטענה:

$k = 0$: במקרה זה $Sol(0) = \{\emptyset\}$

כיוון שלפי ההגדרה $S_i \geq 1$, מחיר הפתרון הריק הוא 0, ולכן :

$$OPT(k=0) = \min_{I \in Sol(k)} \big(w(I)\big) = \min(0) = 0$$

$k > 0$: לפי ההגדרה $OPT(k) = \min_{I \in Sol(k)} \big(w(I)\big)$

לפי טענת העזר $Sol(k) = \big\{Sol(k-i) \cup \{i\}\big\}_{1 \leq i \leq n}$ ולכן :

$$\min_{I \in Sol(k)} \big(w(I)\big) = \min_{I \in \{Sol(k-i) \cup \{i\}\}_{1 \leq i \leq n}} \big(w(I)\big) =$$

$$\min_{I' \in \{Sol(k-i)\}_{1 \leq i \leq n}} (w(i) + w(I')) = \min_{I' \in \{Sol(k-i)\}_{1 \leq i \leq n}} (i^3 + h_k^2 + w(I')) =$$

אבחנה : ערכי הארגומנט תמיד חיוביים, ולכן ניתן לפרקם לביטוי הבא :

$$\min_{I' \in \{Sol(k-i)\}_{1 \leq i \leq n}} (i^3 + h_k^2 + w(I')) \overset{\text{הגדרת OPT}}{=}$$

$$\min_{1 \leq i \leq n} (i^3 + h_k^2 + OPT(k-i)) \blacksquare$$

סעיף ב'

```
אלגוריתם איטרטבי :
נתחזק מערך בגודל n+1 איברים, כאשר בסיום הריצה יתקיים M[n] = OPT(n)
<=
אתחול : לכל i, M[0] = 0, M[i] = infinity
הערה : ישנן שפות תכנות המכילות משתנה גלובלי בשם infinity (כמו javascript p5.js),
      זה נובע מהגדרת האופרטור סדר גודל (בהשוואה גודל infinity תמיד יהיה יותר גדול).

צעד : לכל j = 1,..,n בצע : (התחל ב j=1)
      לכל t = 1,..,n בצע : (התחל ב t=1)
            M[j] = min(M[j], t^3 + h_j^2 + M[j − t])

סיום : החזר את M. ערך הפתרון האופטילי נמצא באיבר האחרון במערך , M[n].
```

.14	**ניתוח זמן ריצה :**
.15	אתחול : אתחול המערך , מעבר על כל האיברים $O(n)$.
.16	צעד : מספר הצעדים הינו כמספר הלולאות המתקיימות, במקרה הגרוע :
.17	$n-1+n-2+..+1 \overset{\text{סדרה חשבונית}}{=} \frac{n\left(2+(n-1)\right)}{2} = \frac{n(n+1)}{2} = O(n^2)$
.18	עלות כל צעד הינה בדיקת min וחישוב מספרי $O(1)$
.19	בסה"כ $O(n^2)$.
.20	
.21	
.22	
.23	
.24	

סעיף ג'

.1	בתום ריצתו של אלגוריתם סעיף ב' מתקיים $M[i] = OPT(i)$ לכל $0 \le i \le n$
.2	**תיאור האלגוריתם :**
3.	Reconstruct (i)
4.	if (i = 0) return ∅;
5.	else
6.	minindex = 0;
7.	for (j= 0 : I)
8.	if $((minindex)^3 + M[i - minindex] > j^3 + M[i-j]$
9.	minindex = j;
10.	return {minindex } ∪ $Reconstruct(i-j)$;
.11	
.12	**הוכחת האלגוריתם :**
.13	לצורך הוכחת הנכונות נניח תמיד כי ההנחה (*) $M[i] = OPT[i]$ מתקיימת לכל i.
.14	**משפט :** תחת ההנהחה האלגוריתם הרקורסיבי משחזר פתרון אופטימלי לבעיה.
.15	**טענת עזר :** תחת ההנחה (*), לכל i הקריאה Reconstrcut(i) מחזירה פתרון
.16	$I \in Sol(i)$ במחיר $M[i]$

הוכחת המשפט על סמך טענת העזר : עבור $i = n$ הקריאה Reconstruct(i) מחזירה פתרון $I \in Sol(n)$ במחיר $M[n]$. I פתרון חוקי, ותחת ההנחה המחיר M[n] הינו אופטימלי $M[n] = OPT(n)$ ■

הוכחת טענת העזר : באינדוקציה על i.

בסיס : $i = 0$, הפתרון היחיד ב $Sol(0)$ הינו הפתרון הריק, והאלגוריתם מחזיר את הקבוצה הריקה.

הנחת האינדוקציה : לכל $j \in \{1,..,n-1\}$ הקריאה $Reconstruct(j-1)$ מחזירה פתרון $I \in Sol(j-1)$ במחיר $M[i-1]$.

צעד : נתבונן בקבוצה המוחזרת מהקריאה $Reconstruct(n)$:

$I = \{j\} \cup Reconstruct(n-j)$ כאשר $j \in \{1,..,n-1\}$ כלשהו.

נשים לב שהנחת האינדוקציה עובדת על $Reconstruct(n-j)$ ונקבל :

$I = \{j\} \cup I' \mid I' \in Sol(n-j)$.

לפי טענת העזר מסעיף א' קבוצת הפתרונות $Sol(n)$ הינה בדיוק קבוצת כל הפתרונות ב $\{\{j\} \cup Sol(n-j)\}_{(1\le j\le n)}$, כלומר $\underline{I \in Sol(n)}$.

האלגוריתם הרקורסיבי בודק את מינימום המחיר בכדי למצוא את גודל הקפיצה האחרונה שנלקחה בהתאם לנוסחת המבנה

$$P(I) = OPT(n) = \min_{1\le i\le n}\left(i^3 + h_n^2 + OPT(n-i)\right)$$

ומההנחה (*) נקבל כי M[n]=OPT(n), נקבל בסה"כ $\underline{P(I) = M[n]}$ כנדרש■

ניתוח זמן ריצה :

בקריאה ל Reconstrct(n) מספר הקריאות הרקורסיביות הינו $O(n)$, עלת כל קריאה (ללא עלות הקריאה הרקורסיבית במהלכה) הינה O(i) ולכן בסה"כ

$Sum(i) = n-1+n-2+..+1 \overset{\text{סדרה חשבונית}}{=} \frac{n(n+1)}{2} = O(n^2)$.

שאלה 2

סעיף א'

1. $sol(k) = \{I = (a_1, a_2, , , , , , , , , a_k) \mid a_i \in (L_i, H_i, N) \cap a_k \neq H \cap (a_i = H_i \rightarrow a_{i+1} = N)$
2. $OPT(k) = \max_{I \in sol(k)} U(I)$
.3
.4 כל פתרון $I \in Sol(j)$ מקיים בדיוק אחד מהשניים :
.5 $a_j \in H\ v\ L$ **או** $a_j \in null$ $0 \leq j < n$
.6 $a_j \in null, L$ $j = n$
.7

סעיף ב'

1.	0	k=0
2.		
3. OPT(k) =	L_1	k=1
4.		
5.	$\max(H_{k-1} + N + OPT(k-2), L_k + OPT(k-1))$	k > 1

סעיף ג'

1. **א.** עבור k=0
2. $OPT(0) = \max_{I \in sol(0)} U(I),\ \ sol(0) = \emptyset \Rightarrow \max\big(U(\emptyset)\big) = 0$.
3. **ב.** עבור k=1
4. מהגדרת sol(k): $a_k \neq H$ כלומר
5. $Sol(k=1) = \{I = a_1 \mid a_1 \in (L_1, N)\} \Rightarrow OPT(k=1) = \max_{I \in sol(I)} U(I) =$
6. $\max(\{L_1\} \vee \{N\})$
7. ומכיוון ש $\lvert N \rvert \geq \lvert L_1 \rvert \Leftarrow OPT(1) = L_1$.
8. **ג.** $k > 1$
9. **טענת עזר :** הפתרונות החוקיים ב $Sol(k)$ המכילים את $a_{k-1} = H_{k-1}$ הינם מהצורה
10. $I \in sol(k-2) \cup \{H_{k-1}\} \cup \{N\}$ אחרת הם מהצורה
11. $I \in sol(k-1) \cup \{a_k\} \mid a_k \in (L_k, N)$
12. **הוכחת הנוסחה בעזרת טענת העזר:**
$13. OPT(k) = \max_{I \in sol(k)} U(I) = \max\left(\max_{I \in sol(k), a_{k-1} = H_{k-1}} U(I), \max_{I \in sol(k), a_{k-1} \neq H_{k-1}} U(I)\right)$
14.= מהטענה $= \max(\max(sol(k-2) \cup \{H_{k-1}\} \cup \{N\}), \max(sol(k-1) \cup \{a_k\}))$
15. $= \max(H_{k-1} + OPT(k-2), OPT(k-1) + L_k)$, $(\lvert N \rvert = 0)$
16. הוכחת טענת העזר :
17. יהי פתרון חוקי I. מהגדרת פתרון חוקי, מתקיימת אחת מ 2 האפשרויות :
18. $a_{k-1} = H_{k-1}$ או $a_{k-1} \neq H_{k-1}$.
19. **עבור** $a_{k-1} = H_{k-1}$: נשים לב שמכיוון שזהו פתרון חוקי, מתקיים $a_k = N$, $a_{k-2} \notin H$
20. ובנוסף $k-2$ האיברים הראשונים ב I מהווים רצף חוקי
21. ולכן $I = I' \cup a_{k-1} \cup Null$ כאשר $I' \in Sol(k-2)$ מהגדרת פתרון חוקי
22. <= $I \in Sol(k-2) \cup a_{k-1} \cup Null$.
23. **עבור** $a_{k-1} \neq H_{k-1}$: נשים לב שמכיוון שזהו פתרון חוקי $k-1$ האיברים
24. הראשונים ב I מהווים רצף חוקי כאשר $a_{k-1} \neq H_{k-1}$
25. ולכן $I = I' \cup a_k$ כאשר $I' \in Sol(k-1)$ ו- $a_k \neq H_k$ מהנחת I פתרון חוקי
26. <= $I \in Sol(k-1) \cup a_k$ ■

סעיף ד'

1.	נתחזק מערך m בגודל n+1 , שבו בכל סיום צעד J m[j] = OPT(j) כאשר j בתחום
2.	**א.** אתחול המערך באיברי 0 ואת האיבר 1 בL_1 כלומר m[1] = L_1
3.	**ב.** עבור j = 2 to n
4.	$m[j] = \max\{H_{j-1} + N + m[j-2], L_j + m[j-1]\}$
5.	**ג.** נחזיר את המערך והערך במקום m[n] יהיה מקסימלי
6.	**הוכחת נכונות האלוגריתם :**
7.	**טענה :** לכל $1 \le j \le n$ מתקיים ש $m[j] = OPT(j)$
8.	**הוכחת האלגוריתם :** אם נקרא לאלגוריתם עם n נקבל m[n] = OPT(n) ■
9.	נוכיח את הטענה באינדוקציה על j
10.	**בסיס :** j=0 אז m[0]=0 כיוון והמערך מאותחל באפסים לכן m[0]=OPT(0) =0
11.	J=1 $m[j] = L_1$ ומנוסחת המבנה $OPT(1) = L_1$ כלומר m[1] = OPT(1)
12.	**הנחת האינדוקציה:** נניח כי עבור $1 \le j-1 \le n$ מתקיים $m[j-1] = OPT(j-1)$
13.	**צעד :** ננתח את שתי האפשרויות:
14.	1. $m[j] = L_j + m[j-1] =$ לפי הנחת אינדוקציה $= L_j + OPT(j-1)$
15.	2. $m[j] = H_{j-1} + N + m[j-2] =$ לפי הנחת אינדוקציה $= H_{j-1} + N + OPT(j-2)$
16.	מ1 ו2 נובע ש
17.	$m[j] = \max(H_{j-1} + N + OPT(j-2), L_j + OPT(j-1))$
18.	ולפי נכונות נוסחת המבנה m[j] = OPT(j) ■
19.	
20.	**ניתוח זמן ריצה :**
21.	1. אתחול המערך בO(n)
22.	2. צעד ב O(1) n-2 צעדים כאלה אזי O(n)
23.	<= לכן סה"כ O(n).
24.	
25.	
26.	
27.	

.28
.29

סעיף ה' - **orenrot** אישר עד 50 שורות בפורום.

1. $Reconstruct(j)\{$
2. $I \quad f(j == 0)\ return\ \emptyset$
3. $I \quad f(j\ ==\ 1)\ return\ L_1$
4. $I \quad f(m[j]\ ==\ H_{j-1} + m[j-2])$
5. $Return\ reconstruct(j-2) \cup \{H_{j-1}\} \cup \{N\}$
6. $R \quad eturn\ Reconstruct(j-1) \cup \{L_j\}$
.7 **הנחה:** נתון מערך M בו מתקיים לכל $0 \leq j \leq n$ כי m[j] = OPT(j)
.8 הוכחת האלגוריתם :
.9 **משפט :**_תחת ההנחה האלגוריתם מחזיר פתרון אופטימלי
.10 **טענה עזר :** תחת ההנחה, הקריאה לפונקציה במקום הj מחזירה $I \in sol(j)$ במחיר
.11 m[j]
.12 **הוכחת המשפט :** בעזרת טענת העזר j=n מחזיר $I \in sol(j)$ במחיר m[j] = OPT(n)
.13 **הוכחת טענת העזר:**_ נוכיח באינדוקציה על J
.14 **בסיס:** j=0 , במקרה הזה $sol(0) = \emptyset$ וm[0]=0
.15 J=1 , במקרה הזה $sol(1) = \{I = (a_1) \vert a_1 \in (L_1, N)$ ומכיוון ו$L_1 \geq N$
.16 $I = L_1$, בדיוק מה שהאלגוריתם מחזיר ו $m[1] = L_1$ (מהאלגוריתם האיטרטיבי)
.17 **הנחת האינדוקציה:**_הקריאה Reconstruct(j-1)_מחזירה $I \in sol(j-1)$_במחיר
.18 M[j-1]
.19 **צעד :**
.20 **א.** האלגוריתם הרקורסיבי מחזיר בשלב ה j אחת מ 2 האפשרויות :
.21 Reconstruct(j-1)$\cup \{L_j\}$ **או** Reconstruct(j-2) $\cup \{H_{j-1}\} \cup \{N\}$
.22 **עבור** Reconstruct(j-1)$\cup \{L_j\}$: נשים לב שמתקיים מהנחת האינדוקציה
.23 האיברים הראשונים שלו מהווים רצף חוקי כאשר $a_{j-1} \notin H$
.24 ולכן $I = I' \cup L_j$ הינו פתרון חוקי עבור j איברים

.25		<= $I \in Sol(j)$ = Reconstruct(j-1)∪ $\{L_j\}$
.26		**עבור** Reconstruct(j-2) ∪ $\{H_{j-1}\} \cup \{N\}$: מהנחת האינדוקציה
.27		
.28		$I' \in sol(j-2)$ = Reconstruct(j-2), מכיוון ש I' פתרון חוקי $j-2$
.29		האיברים הראשונים שלו מהווים רצף חוקי כאשר $a_{j-2} \neq H_{j-2}$
.30		ולכן $I = I' \cup \{H_{j-1}\} \cup \{N\}$ הינו פתרון חוקי עבור j איברים
.31		<= $I \in Sol(j)$ = Reconstruct(j-2) ∪ $\{H_{j-1}\} \cup \{N\}$
.32		בסה"כ האלגוריתם הרקורסיבי מחזיר בשלב ה j פתרון $I \in Sol(j)$.
.33	**ב.**	**עבור** I = Reconstruct(j-2) ∪ $\{H_{j-1}\} \cup \{N\}$: האלגוריתם יחזיר זאת
.34		**אם** מתקיים m[j] == $H_{j-1} + m[j-2]$. לפי הנחת האינדוקציה
.35		Reconstruct(j-2) מחזירה פתרון במחיר $m[j-2]$, ובסה"כ מחיר I
.36		המוחזר הינו H_{j-1} + $m[j-2]$, בדיוק m[j].
.37		**עבור** I = Reconstruct(j-1)∪ $\{L_j\}$: לפי הנחת האינדוקציה
.38		$I' \in sol(j-1)$ = Reconstruct(j-1) במחיר m[j-1].
.39	.	לפי האלגוריתם האיטרטיבי, מתקיים :
.40		$m[j] = \max\{H_{j-1} + N + m[j-2], L_j + m[j-1]\}$
.41		אך מכיוון שאנו באופציה השנייה של האלגוריתם הרקורסיבי אזי בפתרון
.42		האיטרטיבי התקיים $m[j] = L_j + m[j-1]$,
.43		המחיר של I הינו L_j + m[j-1] מהגדרתו, בדיוק המחיר של $m[j]$.
.44		<= בסה"כ האלגוריתם הרקורסיבי מחזיר בשלב ה j פתרון $I \in Sol(j)$ במחיר $m[j]$
		■

שאלה 3

סעיף א'

1.	$Sol(r,k) = \{I: I \subseteq V \mid \quad (\lvert I \cap V_{blue} \rvert = k\ ) \wedge$
2.	$\forall (u,v) \in E \to u \vee v \vee u,v \in I\}$
3.	
4.	$OPT(r,k,V') = \min\limits_{\mathrm{I} \in Sol(r,k)} \lvert I \rvert$
5.	
6.	כל פתרון $I \in Sol(j)$ מקיים בדיוק אחד מהשניים :
7.	$r \in I$ או $r \notin I$
8.	
9.	פונקציית עזר $isBlue(nodes)$ מחזירה את **מספר** הקודקודים הכחולים מ $nodes$.

סעיף ב' – אושר בפורום 15 שורות.

1.	$OPT(r,k,V') =$	
2.	$\min(1 + \min\limits_{0 \le j \le k_1}(OPT(r.right,\ k_1 - j, V_1) + OPT(r.left,\ j, V_1),$	
3.	$2 + \min\limits_{0 \le j \le k_2}(OPT(r.right,\ k_2 - j, V_2) + OPT(r.left,\ j, V_2))$	
4.		$V_1 = V' \cup \{r\}, \quad V_2 = V' \cup \{r.right, r.left\}$
5.		$r \notin V'$
6.		$k_1 = k - isBlue(r),\ \ k_2 = k - isBlue(r.right, r.left)$
7.		
8.	$\min\limits_{0 \le j \le k}(OPT(r.right, k - j, V') + OPT(r.left,\ j, V')$	$r \in V'$
9.		
10.	0	$(r = null \mid\mid r \in V', r.right \,\&\, r.left = null\ ) \wedge k = 0$
11.		
12.	∞	$[k = 0 \wedge\ r \notin V' \wedge (isBlue(r, r.right) = 2 \quad \vee$
13.		$isBlue(r, r.left) = 2)]$
14.		$\vee\ [r.right \,\&\, r.left = null\ \ \wedge k > isBlue(r)$
15.		

סעיף ג'

מקרה בסיס : עבור k=0 ועץ ריק נקבל OPT=0 .

אבחנה : עבור השורש r אם $r \notin V'$ אזי 2 הבנים שלו **יצטרכו** להיות ב V' על-מנת לכסות את הקשתות $(r, r.right), (r, r.left)$ **או** שאנו מצרפים את r ל V', כיסינו את 2 הקשתות הללו ונמשיך לתתי הבעיות עבור בניו.

מקרה א' : אם $r \notin V'$ אזי לפי האבחנה אנו בוחרים ב 2 דרכי פעולה האפשריים, המובעים בפירוק הפתרון הבא :

$$Sol(r,k,V') = \{Sol(r.right, k_1 - j, V_1) \cup Sol(r.left,\ j, V_1) \cup \{r\}\}_{0 \le j \le k_1} \cup \{Sol(r.right, k_2 - j, V_2) \cup Sol(r.left,\ j, V_2) \cup \{r.right, r.left\}\}_{0 \le j \le k_2}$$

$$k_1 = k - isBlue(r),\quad k_2 = k - isBlue(r.right, r.left)$$

$$V_1 = V' \cup \{r\},\quad V_2 = V' \cup \{r.right, r.left\}$$

מהגדרת OPT הגודל המינימלי של הכיסוי, נבצע min על קבוצת הפתרונות הללו ונקבל את החלק הראשון בנוסחאת ה OPT.

מקרה ב' : אם $r \in V'$ אזי הקשתות לבניו מכוסות. נעביר את הבעיה לתתי בעיות עבור בניו לכסות את תתי העצים + כיסויים המכילים k קודקודים כחולים

$$Sol(r,k,V') = \{Sol(r.right, k - j, V_1) \cup Sol(r.left,\ j, V_1)\}_{0 \le j \le k}$$

מקרה ∞ : 1. אם $r \notin V'$, אנו צריכים לכסות את הקשתות היוצאות לבניו. במידה ו r כחול וגם אחד מבניו כחול, ואנו נדרשים לספק כיסוי המכיל k=0 קודקודים כחולים – מקרה זה לא פתיר ולכן נחזיר ∞.

2. אם $r \notin V'$, r הינו עלה $(r.right, r.left = null)$ ואנו צריכים לספק מתת העץ שלו(כולל אותו) k=1 קודקודים כחולים כאשר r אינו כחול – מקרה זה לא פתיר ונחזיר ∞.

תוספת למקרה 0 : אם $r \in V'$, r הינו עלה $(r.right, r.left = null)$ ואנו נדרשים לספק כיסוי המכיל 0 קודקודים כחולים אזי נחזיר 0 (הפתרון הריק).

סעיף ד'

מבנה נתונים : לכל צומת בעץ בינארי מושלם T יהיו 2 מערכים בגודל k, A_1, A_2. התא $v.\mathrm{A}_1[i]$ מתאר את ערך
הפתרון עבור כיסוי המכיל i קודקודים כחולים בתת העץ של v, הכיסוי יכול להכיל את
הצומת. התא $v.\mathrm{A}_2[i]$ מתאר את ערך הפתרון עבור כיסוי המכיל i קודקודים כחולים בתת
העץ של v, הכיסוי לא יכול להכיל את הצומת.
אלגוריתם איטרטיבי :
1. אתחול : לכל $v \in V$ $\mathrm{A}_{1,2}[i] = \infty$ $\forall 0 \le \mathrm{i} \le \mathrm{k}$
2. מיין את V לפי $post - order$, קודקודים בנים יופיעו לפני אבות.
3. לכל $r \in V$ בצע :
3.1 אם $r.right = r.left = null$ בצע את האתחול הבא :
3.1.1 אם $isBlue(r) = 1$: $\mathrm{A}_1[1] = 1,\ A_1[0] = 0,\ A_2[0] = 0$
3.1.2 אחרת : $A_1[0] = 0,\ A_2[0] = 0$
3.1.3 break
3.2 אם $r = \text{root of T}$:
$return\ \min(r.right.A_1[k-i] + r.left.A_1[i]\ ,\ 1 + r.right.A_1[k_1 - i] + r.left.A_1[i],$
$1 + r.right.A_2[k_1 - i] + r.left.A_2[i]) \quad \| \ k_1 = k - isBlue(r)$
3.2 אחרת לכל $j\ from\ 0\ to\ k$:
3.2.1 אם $(isBlue(r, r.right) = 2\ \|\| isBlue(r, r.left) = 2) \wedge\ j = 0$: $break$
3.2.2 $k_1 = k - isBlue(r),\ k_2 = k - isBlue(r.right, r.left), min1 = \infty, min2 = \infty$
3.2.3 אם $\mathrm{k}_1 \ge 0$, לכל $i\ from\ 0\ to\ j$:
$min1 = \min(min1,\ 1 + r.right.A_1[k_1 - i] + r.left.A_1[i])$
3.2.4 אם $\mathrm{k}_2 \ge 0$, לכל $i\ from\ 0\ to\ j$:
$min2 = \min(min2,\ \ 2 + r.right.A_2[k_2 - i] + r.left.A_2[i])$
3.2.5 $r.A_1[j] = min1,\ r.A_2[j] = min2$
ניתוח זמן ריצה : post-order עולה בעץ בינארי מושלם $\mathrm{O}(\|V\|)$
עבור כל עלה האלגוריתם יעלה $\mathrm{O}(1)$, סה"כ עלים $\log(\|V\|)$
עבור שאר הצמתים מספר הלולאות יהיה $1 + 2 + .. + k \overset{\text{סדרה חשבונית}}{=} \frac{k(k+1)}{2} = O(k^2)$
עלות כל לולאה $\mathrm{O}(1)$. אתחול : כמספר תאי כל המערכים $2 * \|\mathrm{V}\| * k$ $\mathrm{O}(\|\mathrm{V}\| * \mathrm{k})$
סה"כ זמן ריצה $\mathrm{O}(n) + O(\|V\| * k) + O(\log(\|V\|)) + O(k^2) => O(k^2 + \|V\| * k)$

סעיף ה'

1.	$\boldsymbol{Reconstruct}\ (\boldsymbol{r}, \boldsymbol{k}, \boldsymbol{V'})$ \\ we call this function with (T.root, k, ∅)
2.	$if\ (k = 0\ \&\&\ r = null)\ \ \boldsymbol{return}\ \emptyset;$
3.	$if\ (k = 0\ \&\&\ r \notin V' \&\&\ ((isBlue(r, r.right) = 2\ \ \|\|isBlue(r, r.left) = 2\ \|\|$
4.	$r.right = r.left = null)\ \|\|\ (r.right = r.left = null\ \&\&\ k > isBlue(r))$
5.	$\boldsymbol{return}\ \{-1\};$ \\ a sign for not possible, like infinity
6.	$if\big(r.right = r.left = null\ \&\ k = isBlue(r)\big)\quad \boldsymbol{return}\ \{r\} \cup V';$
7.	$if\ (r \in V')\quad k_1 = k,\quad V_1 = V'\ , min1 = \{-1\}\ ;$
8.	$if\ (r \notin V')\quad k_1 = k - isBlue(r),\quad V_1 = V' \cup \{r\}\ , min1 = min2 = \ \{-1\};$
9.	$for\ (j{:}\ 0\ to\ k_1)$
10.	$A = OPT(r.right, k_1 - j, V_1)$
11.	$B = OPT(r.left, j, V_1)$
	$if\ ((A\ , B \neq \{-1\})\&\&\ (\text{min1} = \{-1\}\ \|\|(\text{min1} \neq \{-1\}\&\&\ \|\ \text{min1}\| > \|A \cup B\|)))\ \ \text{min1} = A \cup B;$
	$if\ (r \in V')$
12.	$if\ min1 = \{-1\}\ \ \boldsymbol{return}\ min;\quad else\ \ \boldsymbol{return}\ V' \cup min1;$
13.	$if\ (r \notin V')\quad k_2 = k - isBlue(r.right, r.left),\quad V_2 = V' \cup \{r.right, r.left\}$
14.	$for\ (j{:}\ 0\ to\ k_2)$
15.	$A = OPT(r.right, k_2 - j, V_2)$
16.	$B = OPT(r.left, j, V_2)$
	$if\ ((A\ , B \neq \{-1\})\&\&\ (\text{min2} = \{-1\}\ \|\|(\text{min2} \neq \{-1\}\&\&\ \|\ \text{min 2}\| > \|A \cup B\|)))\ \ \text{min2} = A \cup B;$
17.	$if\ min1, min2 = \{-1\}\quad \boldsymbol{return}\ \{-1\};$
18.	$else\ if\ \ (min1 = \{-1\})\quad \boldsymbol{return}\ min2 \cup V';$
19.	$else\ if\ \ (min2 = \{-1\})\quad \boldsymbol{return}\ min1 \cup V';$
20.	$else\ if\ \ (\|min1\| > \|min2\|)\quad \boldsymbol{return}\ min2 \cup V';$
21.	$else\ \boldsymbol{return}\ min1 \cup V';$
22.	**ניתוח זמן ריצה :** מספר הקריאות הרוקריסיות הינן $\text{O}(\|V\| * k)$
23.	עלות כל קריאה רקוריסיבית (ללא עלות הקריאות שבתוכה) הינה $O(k)$ ולכן בסה"כ $\text{O}(\|V\| * k^2)$

סעיף ו'

24.	**אבחנה** : מבנה הנתונים מסעיף ד' מכיל מידע עבור כל קומבינציות k שתדרש מהבנים
25.	**1.** ולכן עבור השורש של T נוכל להוסיף לאלגוריתם האיטרטיבי :
26.	$min1 = \infty, mink = 0, temp;$
27.	$for\ (\ j\ :\ 0\ to\ k)$
28.	$temp = min1;$
29.	$for\ (\ i\ :\ 0\ to\ j)$

$j_1 = j - isBlue(r);$

$min1 = \min(min1, r.right.A_1[j - i] + r.left.A_1[i],$

$1 + r.right.A_1[j_1 - i] + r.left.A_1[i],$

$1 + r.right.A_2[j_1 - i] + r.left.A_2[i])$

$if\ min1\ \neq tmep \rightarrow mink = j;$

אבחנה : כעת אנו יודעים מהו גודל הקבוצה המינימלית $min1$ וכמה קודקודים כחולים
היא מכילה $mink$.

2. כעת נקרא לאלגוריתם הרקורסיבי עבור הפרמטרים $(T.root, mink, \emptyset)$ ונקבל
כתשובה V' כך ש $|V' \cap V_{blue}| = mink \le k$, V' כיסוי בעל גודל מינימלי.

ניתוח זמן ריצה : הרצת האלגוריתם האיטרטיבי $O(k^2 + |V| * k)$
שימוש במבנה הנתונים $O(k^2)$, שימוש באלגוריתם
הרקורסיבי $O(k^2 * |V|)$ ובסה"כ $O(|V|(k^2 + k))$.

שאלה 4

סעיף א'

1. אלגוריתם גנרי בודק אם קיימת קשת (u,v) עבורה מתקיים $d(v) > d(u) + w(u,v)$.
2. נניח כי קיימת קשת (u,v) המקיימת $d(v) > d(u) + w(u,v)$.

4. **אבחנה :** תהי $(u,v) \in E$ אזי מתקיים $\delta(s,v) \le \delta(s,u) + w(u,v)$
5. כאשר $\delta(s,v)$ – מסלול קל ביותר מ s ל u.
6. הביטוי הימני באי השוויון מייצג משקל אחד המסלולים מ s ל v ולא בהכרח הקל ביותר.

8. **טענת עזר :** יהי $P_{1k} =< V_1,..,V_k >$ מסלול קל ביותר מ V_1 ל V_k ב G. לכל 2

.9	קודקודים $V_i, V_j \ni P_{1k}$ המסלול ביניהם ב P_{1k} הינו המסלול הקל ביותר.
.10	
.11	**הוכחת טענת העזר :** נניח כי P_{1k} מסלול קל ביותר, אך קיימים V_i, V_j כך ש
.12	$< V_i, .., V_j > \subseteq P_{1k}$ איננו המסלול הקל ביותר. אזי קיים P'_{ij} המהווה מסלול
.13	קל ביותר, ונוכל להרכיב מסלול P'_{1k} המקיים :
.14	$< V_1, .. > \cup P'_{ij} \cup <, .. V_k > < P_{1k}$ בסתירה להנחה ש P_{1k} מסלול קל
.15	ביותר.
.16	
.17	<= נסתכל על הצומת $v' = v$. נסמן את המסלול הקצר ביותר
.18	$P_{sv} =< s, P_1, .., P_n, u', v >$. המסלול הקצר ביותר מקיים :
.19	$\delta(s,v) = \delta(s,u') + \delta(u',v) = \delta(s,u') + w(u',v)$
.20	
.21	נשים לב כי מתקיים מהגדרת מסלול-קל-ביותר לכל קודקוד $p \in P_{1k}$ עם צומת (p,v):
.22	$d(p) + w(p,v) \leq \delta(s,u') + w(u',v)$
.23	ובפרט עבור הקודקוד u, אם כך :
.24	$d(v) > d(u) + w(u,v) \geq \delta(s,u') + w(u',v) = \delta(s,u)$
.25	בסה"כ נקבל $d(v) \neq \delta(s,u)$ ■
.26	

סעיף ב'

.1	נניח כי האתחול של האלגוריתם הינו האתחול של האלגוריתם הגנרי :
.2	$d[s] = 0, \forall v \neq s \in V\ d[v] = infinity$
.3	נניח באינדוקציה כי המסלול הקצר ביותר בין s ל v כלשהו הינו $P_{sv} =< s, .., u, v >$,
.4	$(u,v) \in E$.
.5	
.6	**הנחת האינדוקציה :** לפי המיון הטופולוגי כאשר מגיעים ל u כבר מתקיים
.7	$d[u] = \delta(s,u)$
.8	
.9	**צעד :** כשמגיעים ל u מתבצע $relax(u,v,w)$ ומתקיים לאחר מכן :
.10	$d(v) \leq d(u) + w(u,v) \quad \rightarrow \quad d(v) \leq \delta(s,u) + w(v,u)$

.11	אגף ימין הוא תת מסלול של המסלול הקצר-ביותר + אורך קשת (u, v)
.12	לכן בסה"כ אגף ימין שווה ל $\delta(s, u) + w(v, u) = \delta(s, v)$
.13	מכיוון שתמיד מההגדרה $d(v) \geq \delta(s, v)$ אזי $d(v) = \delta(s, v)$
.14	
.15	
.16	
.17	
.18	