תרגול 5 – תכנון דינאמי

עידו עומד מול בניין בן 100 קומות. יש לו 2 מנורות זהות שמכילות ג'ייני כל אחת. אם הוא זורק מנורה מהקומה הנכונה, היא נפתחת, הג'ייני יוצא לחופשי, והוא יכול להביע משאלה.

אם הוא זורק מנורה מקומה נמוכה מדי, לא קורה לה כלום, והוא יכול לזרוק אותה שוב. אם הוא זורק מנורה מקומה גבוהה מדי, המנורה נשברת (ואי אפשר להשתמש בה שוב), והג'ייני מתעצבן ולא מקיים משאלות.

לכמה זריקות זקוק עידו לפני שיוכל לבקש משאלה?

חידה יותר קלה

- **שאלה**: נניח שיש לנו מנורה אחת ומאה קומות. כמה זריקות נצטרך?

תשובה לחידה עם 2 מנורות

14

הסבר

פתרון: 100. נשתמש רק במנורה 1.

נסיון ראשון לשפר: נזרוק את המנורה הראשונה מקומה 50. נצטרך לכל היותר 50 זריקות (אם היא נשברת).

נסיון שני לשפר: נזרוק את המנורה הראשונה מקומה 10. אם הוא נשברת, נצטרך 10 זריקות. אחרת, נזרוק מהקומה ה-20. אם היא נשברת, נצטרך 11 זריקות וכן הלאה. סה"כ נצטרך 19 זריקות.

נסיון שלישי לשפר: נזרוק את המנורה הראשונה מקומה 14. אם היא נשברת, נצטרך 14 זריקות. אחרת, נזרוק מהקומה ה-27. היא נשברת, נצטרך 14 זריקות וכן הלאה.

תכנות דינאמי

הפתרון לשאלות תכנות דינאמי מורכב ממספר שלבים קבועים:

1) הגדרת תתי בעיות,
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה,
3) חישוב תתי הבעיות הקטנות ביותר,
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן,
5) ניתוח זמן הריצה.

תרגיל 1: יופי של עודף

- **תרגיל:** נתונים n מטבעות מסוג שקלים מאוד חדשים (שמ"ח) בשווי

$$1 = a_1 < a_2 < a_3 < \ldots < a_n$$

אתם מוכרים בלון ב 4 שמ"ח, וקיבלתם שטר של $x \leq 100n^3$ שמ"ח. תארו אלגוריתם שיחשב את מספר המטבעות המינימאלי שתצטרכו בשביל להחזיר עודף מדויק.

דוגמא

המטבעות הן: 1,5,7

$x = 14$

צריך להחזיר 10שמ"ח עודף.

האפשרויות הן: 7,1,1,1

5,5

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

- נגדיר את $c[k]$ להיות המספר המטבעות המינימאלי עבור עודף ששוויו k.

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

הפתרון הוא $c[x-4]$

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

$$c[0] = 0$$

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

$$c[k] = \min_{1 \le i \le n} \{ c[k - a_i] + 1 \}$$

נצטרך להוסיף: $\forall x < 0, c[x] = \infty$

חישוב של $c[k] = \min_{1 \le i \le n} \{c[k - a_i] + 1\}$

לוקח $O(n)$.

נחשב $O(n^3)$ ערכים – סה"כ $O(n^4)$.

בעיית המנורות הכללית

- אנחנו רוצים לדעת מאיזו קומה נפתחת מנורה כשזורקים אותה מבניין בגובה n קומות. לרשותנו m מנורות. כמה זריקות נצטרך לכל היותר בכדי לדעת בוודאות באיזו קומה מנורה נפתחת?

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

נסמן ב- $c(k,j)$ את מספר הזריקות שנצטרך עבור k מנורות ו- j קומות.

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

הפתרון הוא $c(m,n)$

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

$$\forall k : c(k,1) = 1$$

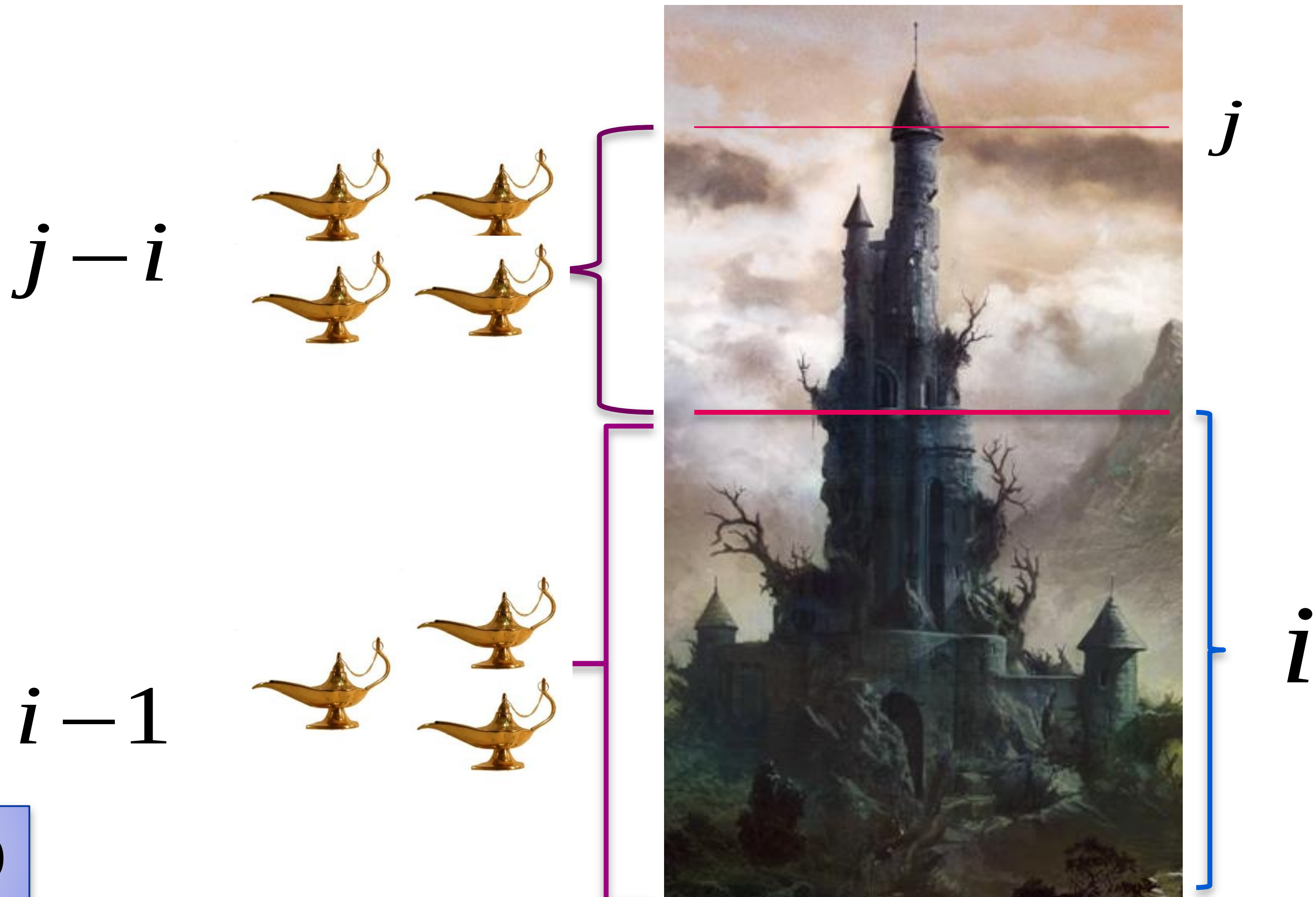
$$\forall j : c(1,j) = j$$

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

$$c(k,j) = \min_i\{\max[c(k,j-i), c(k-1,i-1)]\}+1$$

הסבר בתמונה

$$c(k, j) = \min_i \{\max[c(k, j-i), c(k-1, i-1)]\}+1$$



$(k=4)$

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

נסמן ב- $c(k, j)$ את מספר הזריקות שנצטרך עבור k מנורות ו- j קומות.

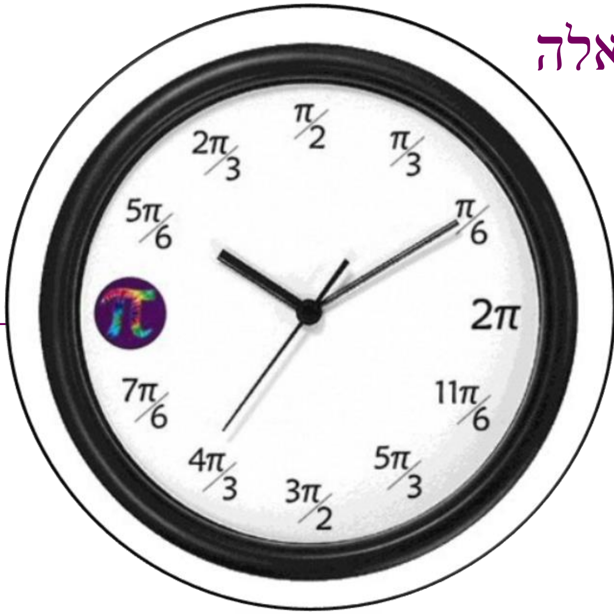
הפתרון הוא $c(m, n)$

$$\forall k : c(k,1) = 1$$

$$\forall j : c(1, j) = j$$

$$c(k, j) = \min_i \{\max[c(k, j-i), c(k-1, i-1)]\} + 1$$

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה



- נחשב $c(k, j)$ לכל k, j :
  - סה"כ mn חישובים.
- כל חישוב הוא:

$$c(k, j) = \min_i \{\max[c(k, j-i), c(k-1, i-1)]\} + 1$$

- לכל היותר n השוואות של 2 ערכים שכבר חישבנו, $O(n)$
וחישוב מינימום על n ערכים:
- סה"כ: $O(mn^2)$

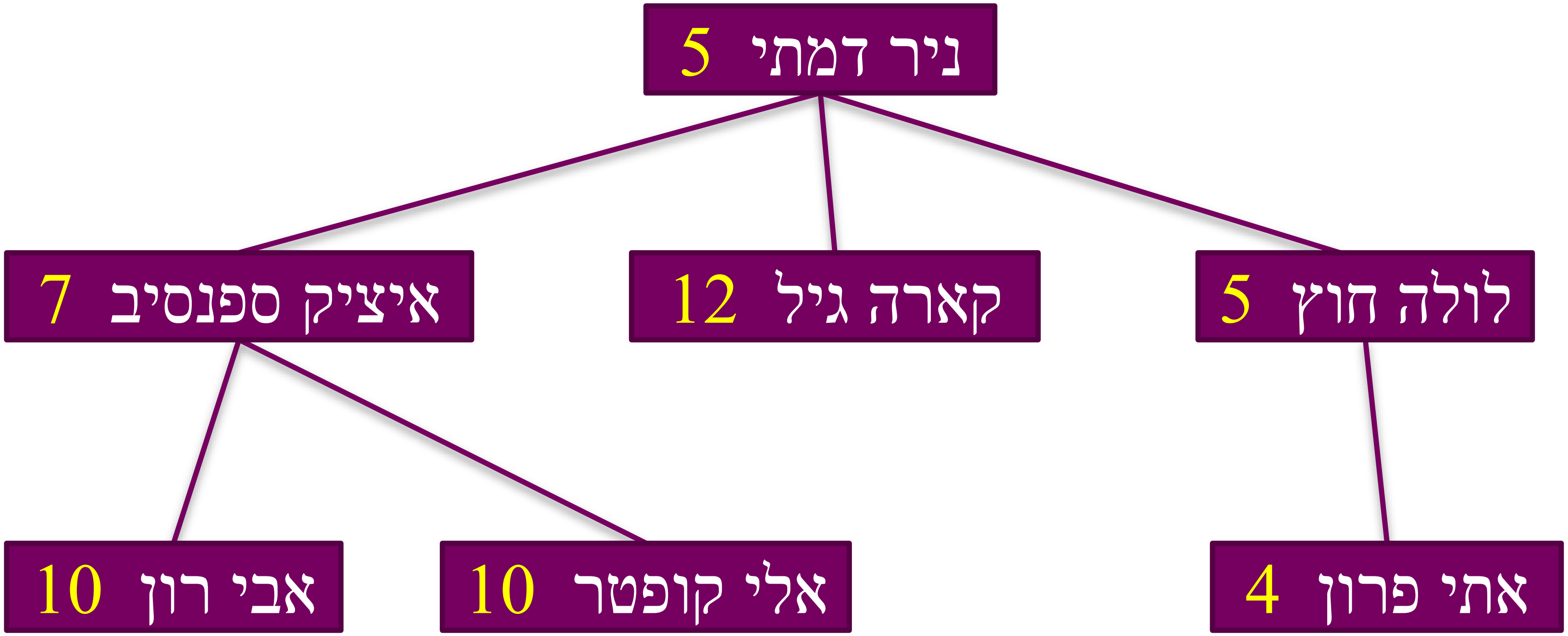
תרגיל 2 : מסיבה

- **תרגיל:** ועדת הקישוט של חברה מארגנת מסיבה, ומנסה להחליט את מי להזמין. בידי הועדה מאגר אשר מחזיק לכל עובד את רמת ה"כיפיות" שלו. בנוסף, ידוע שעובד אינו כיפי כאשר הבוס הישיר שלו נמצא במסיבה, ולכן הוחלט לא להזמין שני אנשים שאחד מהם הינו הבוס הישיר של השני. לצורך כך, הועדה מחזיקה בעץ המתאר את יחסי העובד-בוס בחברה.

תארו אלגוריתם למציאת תת-קבוצת עובדים שממקסמת את רמת הכיפיות במסיבה.

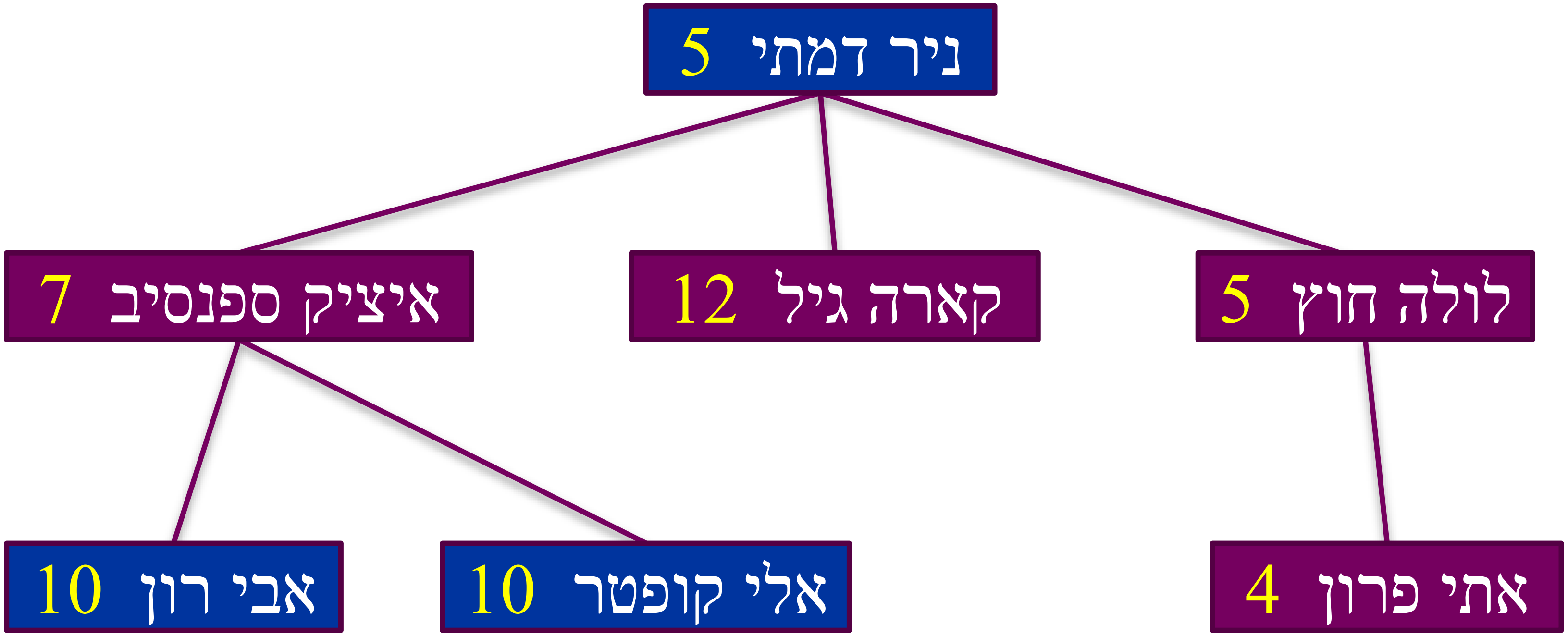
נתחיל בלתאר אלגוריתם שמוצא מהי רמת הכיפיות המקסימלית

מסיבה



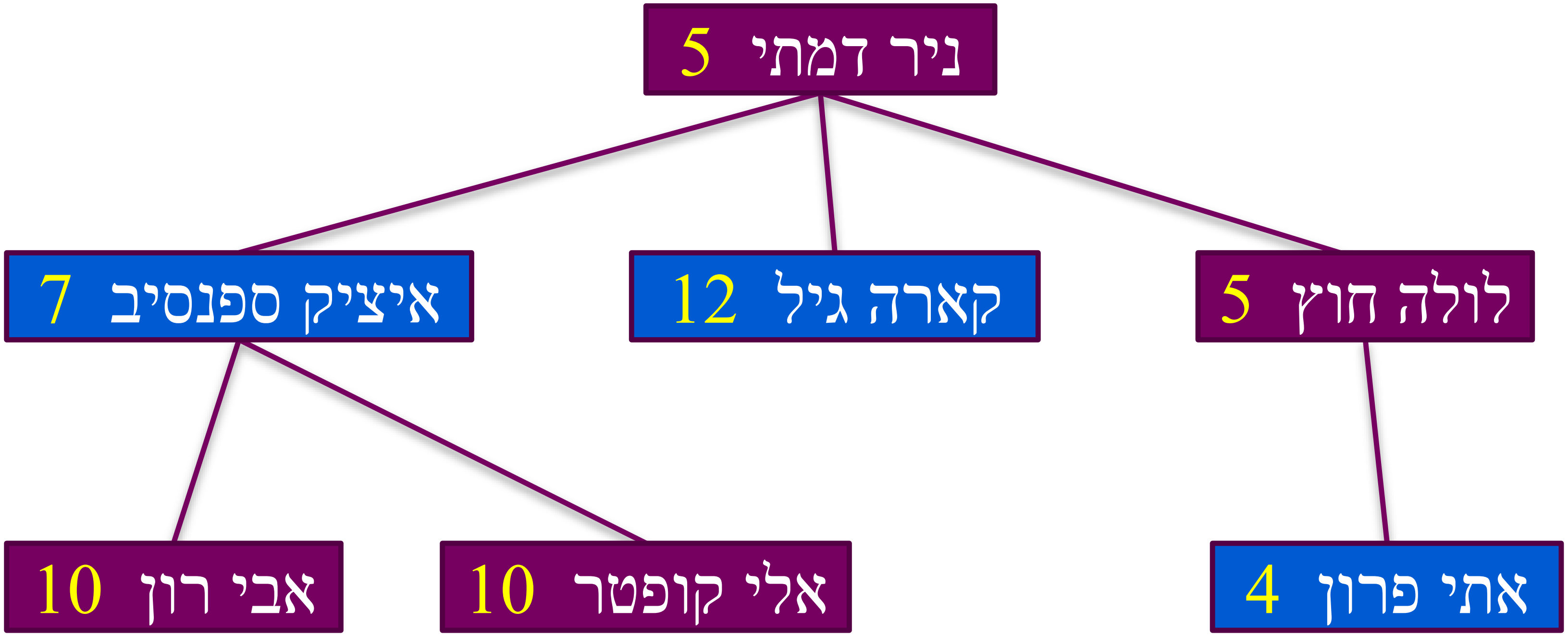
ניר דמתי 5
איציק ספנסיב 7
קארה גיל 12
לולה חוץ 5
אבי רון 10
אלי קופטר 10
אתי פרון 4

דוגמא



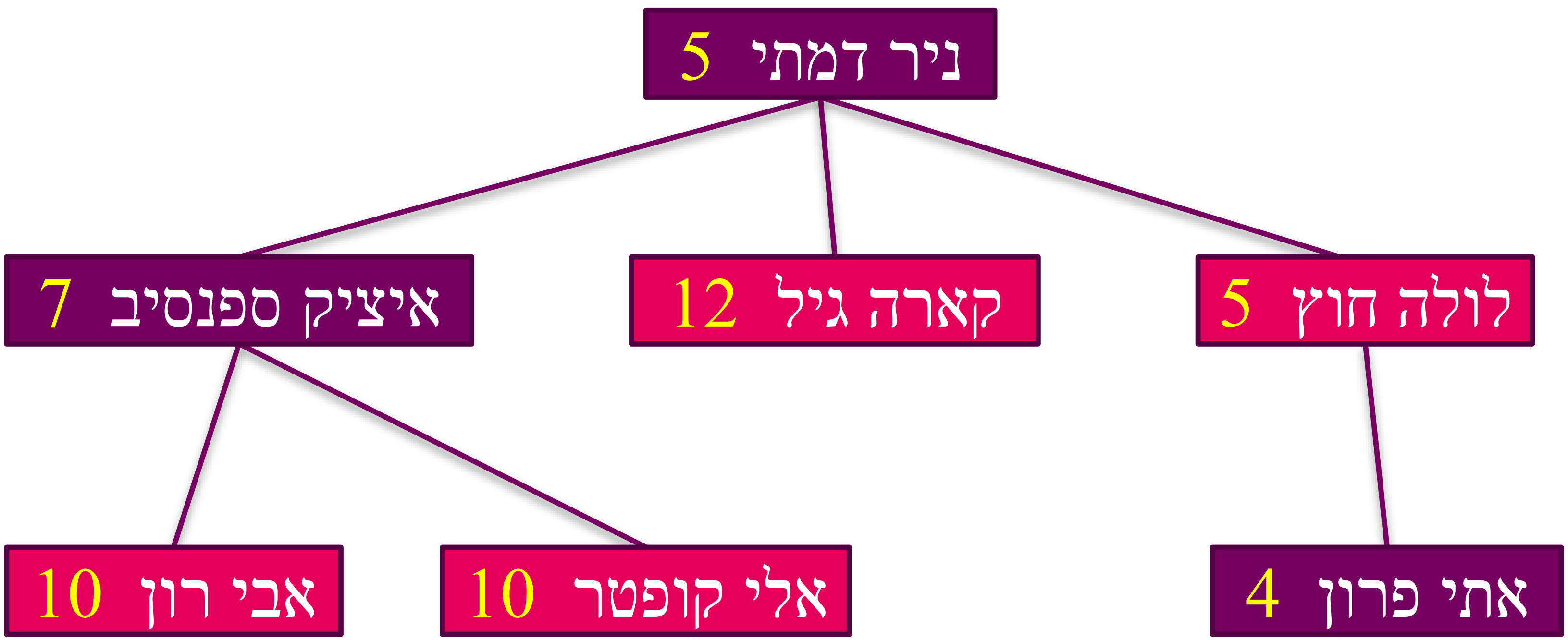
- רמת כייפיות - 25

דוגמא



- רמת כייפיות - 23

דוגמא



- רמת כייפיות - 37

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

- עבור צומת v בעץ, נגדיר את $C(v)$ להיות רמת הכיפיות המקסימאלית במסיבה שבה תת-העץ ששורשו v הוא עץ המוזמנים האפשריים.

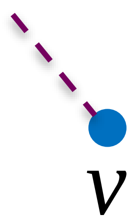
1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

- נסמן את שורש העץ כ- r. אנחנו בעצם רוצים למצוא את $C(r)$.

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

- נסמן את רמת הכיפיות של עובד v כ- $fun(v)$.

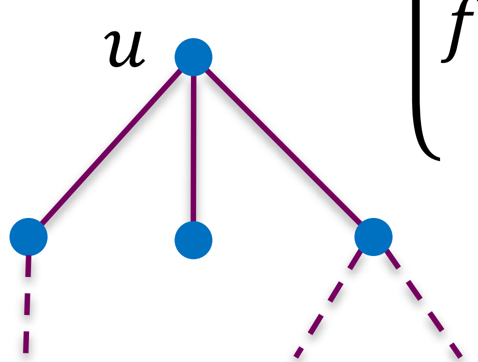
- אם v עלה בעץ:
  - $C(v) = fun(v)$



1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

- נסמן ב-$child(u)$ את קבוצת הבנים (הישירים) של u בעץ.
- אם u איננו עלה בעץ:

$$C(u) = \max \begin{cases} \sum\limits_{v \in child(u)} C(v) \\ fun(u) + \sum\limits_{v \in child(u)} \sum\limits_{w \in child(v)} C(w) \end{cases}$$



סיבוכיות

- נסמן את מספר העובדים בחברה בתור n.
- עבור כל עובד, האלג' מחשב שני ערכים. חישוב של ערך יכול לדרוש חישוב סכום של $O(n)$ איברים.
- כיוון שכל עובד מופיע רק בשני סכומים (בחישוב הערכים של אבא שלו), סך הפעולות שידרשו כל הסכומים יחד יהיה גם הוא $O(n)$.
- לכן, סיבוכיות הזמן של האלג' הינה $O(n)$.

בעיית הסוכן הנוסע

- סוכן רוצה לעבור ב n ערים, פעם אחת בדיוק בכל עיר, ולחזור לאותה העיר. המרחק מעיר i לעיר j הוא d_{ij}.

מה הוא המעבר בין הערים עם מינימום מרחק?

בעיית הסוכן הנוסע

- סוכן רוצה לעבור ב n ערים, פעם אחת בדיוק בכל עיר, ולחזור לאותה העיר. המרחק מעיר i לעיר j הוא d_{ij}.

מה הוא מינימום המרחק הנדרש כדי לעבור בכל הערים?

פתרון נאיבי: לעבור על כל $(n-1)!$ הסדרים האפשריים.

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

נקבע עיר התחלה (1).

נסמן ב $c(A,k)$ עבור $A \subseteq \{2, \dots, n\}$ את מחיר המסלול הקל ביותר מ- 1 ל- k שעובר דרך כל הערים ב A.

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

מי המעגל הקל ביותר?

אם היינו יודעים שהמעגל הקל ביותר נגמר ב k?

ערך הפתרון היה:

$$d_{1k} + c(\{2, \dots, n\} \setminus \{k\}, k)$$

נעבור על כל ה k האפשריים, הפתרון הוא:

$$\min_k d_{1k} + c(\{2, \dots n\} \setminus \{k\}, k)$$

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

$$c(\emptyset, k) = d_{1k}$$

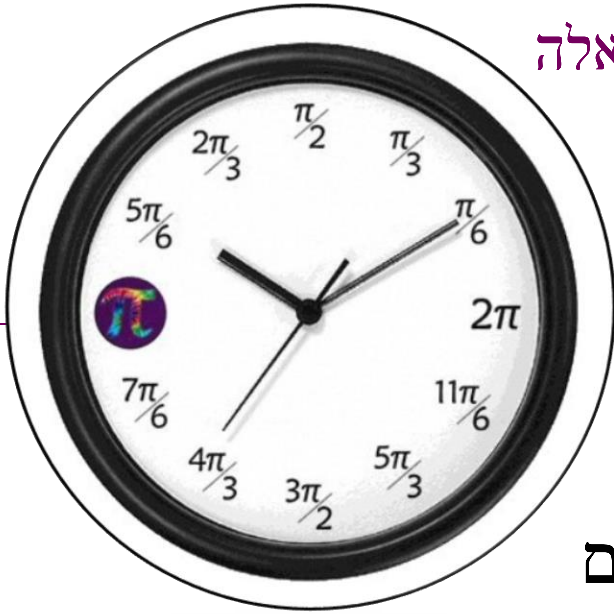
כדי לחשב את $c(A,k)$, אם היינו יודעים מי העיר האחרונה במסלול? נניח שהיא i

$$c(A,k) = d_{ik} + c(A \setminus \{i\}, i)$$

כיוון שאנו לא יודעים מי העיר האחרונה:

$$c(A,k) = \min_{i \in A} c(A \setminus \{i\}, i) + d_{ik}$$

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה



כדי לחשב את $c(A,k)$ צריך לחשב מינימום על לכל היותר n ערכים

סה"כ יש $n \cdot 2^{n-1}$ ערכים לחשב.

לכן סה"כ זמן הריצה הוא $O(n^2 \cdot 2^n)$

הערה: הבעיה הזו היא קשה – לא ידוע על אלגוריתם שרץ בזמן פולינומי.

תרגיל 3 : יופי של סידור

- **תרגיל:** נתונה פסקה המכילה n מילים שאורכן $l_1, l_2, \ldots, l_n$ בהתאמה. רוצים לכתוב את הפסקה על גבי דף המסוגל להכיל m תווים בשורה. בין מילים יש בדיוק רווח אחד, אך ייתכנו רווחים גדולים יותר בסופי שורות. נסמן את מספר הרווחים בסוף השורה ה- i כ- s_i, ונגדיר, עבור סידור המילים ב - k שורות,

$$.ugliness = \sum_{i=1}^{k-1} s_i^2$$

ככל שה – ugliness יותר גבוה, הפסקה נחשבת פחות נעימה לעין. תארו אלגוריתם שמוצא סידור של המילים בשורות אשר מביא ל- ugliness מינימאלי.

דוגמא

עד כמה יפה המשפט הזה: "עד כמה יפה המשפט הזה יכול להיות?" יכול להיות?

דוגמא

- נתון המשפט "עד כמה יפה המשפט הזה יכול להיות?", ואורך שורה $m=10$.

עד.כמה.יפה המשפט..... הזה.יכול.. להיות?	עד.כמה.יפה המשפט.הזה. יכול...... להיות?	עד.כמה.... יפה.המשפט. הזה.יכול.. להיות?
$ugliness =$ $5^2+2^2=29$	$ugliness =$ $1^2+6^2=37$	$ugliness =$ $4^2+1^2+2^2=21$ $=$ *מהמם*!

פתרון

- נגדיר את מספר הרווחים בסוף שורה אשר מתחילה במילה ה- i ומסתיימת במילה ה-j :

$$s(i,j)=m-\sum_{k=i}^{j} l_k-(j-i)$$

- נגדיר עלות שורה שמתחילה במילה ה- i ומסתיימת במילה ה- j :

$$lc(i,j)=\begin{cases}\infty, & s(i,j)<0\\ 0, & j=n\\ s(i,j)^2, & \text{אחרת}\end{cases}$$

דוגמא

$$s(i,j) = m - \sum_{k=i}^{j} l_k - (j-i)$$

$$lc(i,j) = \begin{cases} \infty, & s(i,j) < 0 \\ 0, & j = n \\ s(i,j)^2, & \text{אחרת} \end{cases}$$

$m = 10$

עד כמה יפה המשפט הזה יכול להיות?

1 2 3 4 5 6 7

$s(5,6) =$

$lc(5,6) =$

$s(7,7) =$

$lc(7,7) =$

$s(2,4) =$

$lc(2,4) =$

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

- נגדיר את $c[j]$ להיות ה-ugliness המינימאלי לפסקה אשר מורכבת מ- j המילים הראשונות.

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

הפתרון הוא $c[n]$

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

$$c[1] = lc(1,1)$$

$$lc(i,j) = \begin{cases} \infty, & s(i,j) < 0 \\ 0, & j = n \\ s(i,j)^2, & \text{אחרת} \end{cases}$$

1) הגדרת תתי בעיות
2) מציאת הקשר בין תתי הבעיות שהגדרנו לפתרון השאלה
3) חישוב תתי הבעיות הקטנות ביותר
4) חישוב תתי בעיות באמצעות תתי בעיות קטנות מהן
5) ניתוח זמן הריצה

$$c[j] = \min_{1 \le i \le j} \{ c[i-1] + lc(i, j) \}$$

אילו מילים אנחנו שמים בשורה האחרונה?

סיבוכיות

- נרצה לחשב n ערכים של $c[j]$. לצורך כך, נצטרך לחשב $O(n^2)$ ערכים של $lc(i,j)$ ושל $s(i,j)$.
- חישוב ערך של $s(i,j)$ לוקח זמן קבוע אם משתמשים בערך של $s(i,j-1)$. חישוב $lc(i,j)$ זמן קבוע.
- חישוב ערך של $c[j]$ לוקח $O(n)$ זמן.
- לסיכום, סיבוכיות הזמן של האלג' היא $O(n^2)$.

שיפור האלגוריתם

- שורה יכולה להכיל לכל היותר $m/2$ מילים. לכן:
  - אנו יכולים להניח שמתקיים $lc(i,j)=\infty$ כאשר $j-i>m/2$.
  - נצטרך לחשב רק $O(nm)$ ערכי $lc(i,j)$.
  - אנו יכולים לשנות את ההגדרה של $c[j]$ ל-

$$c[j]=\min_{j-m/2\le i\le j}\{c[i-1]+lc(i,j)\}.$$

- חישוב של $c[j]$ יעלה $O(m)$.
- השינויים הנ"ל מביאים לזמן ריצה $O(nm)$.