שעור 3 משחק מספרים

נתונה סדרה (מערך) של **N** מספרים שלמים חיובים. **N** – מספר זוגי.
במשחק זה משתתפים שני שחקנים. כל אחד מהשחקנים בתורו יכול לבחור מספר אחד מקצה השמאלי או מקצה הימני של הסדרה. מנצח מי שמקבל סכום המספרים שהוא בחר בהם גדול יותר.

דוגמה: נתונה סדרה: 5, 1, 2, 6, 4, 3. שחקן ראשון מתחיל ומנצח.

B	A	step
3	5	1
4	1	2
2	6	3
9	**12**	summa

אסטרטגיה חמדנית.

אלגוריתם חמדן (**Greedy Algorithm**) הוא אלגוריתם המתבסס על היוריסטיקה לפיה בוחרים את האפשרות הטובה ביותר הנראית לעין בשלב הנוכחי, מבלי לקחת בחשבון את ההשפעה של צעד זה על המשך הפתרון.

נתבונן בסדרה: 5, 20, 10, 1. לפי אלגוריתם חמדני:

B	A	step
a_2=20	a_1= 5	1
a_4=1	a_3=10	2
21	**15**	summa

כמו שרואים, האסטרטגיה החמדנית לא מביאה את A לניצחון.

אסטרטגיה של שחקן A שתמיד מביאה אותו לניצחון.

בהינתן סדרת מספרים $a_1, a_2, \ldots, a_n$ (n – מספר זוגי) **A** מחשב שני סכומים: סכום איברים הנמצאים במקומות זוגיים וסכום המספרים הנמצאים במקומות אי-זוגיים: $S_1 = a_1 + a_3 + \ldots + a_{n-1}$, $S_2 = a_2 + a_4 + \ldots + a_n$. במקרה שבו $S_1 \geq S_2$ השחקן A בוחר ב- a_1 ממשיך לבחור במספרים הנמצאים במקומות אי-זוגיים ומנצח.

נתבונן במקרה שבו $S_1 \geq S_2$. מקרה $S_1 \leq S_2$ סימטרי לחלוטין.

נוכיח באינדוקציה כי ל-A יש אפשרות במשך כל המשחק לקחת מספרים הנמצאים במקומות אי-זוגיים וכתוצאה לקבל סכום מספרים גדול יותר.

<u>בסיס האינדוקציה:</u>

כאשר A בוחר ב-a_1 ל-B יש שתי אפשרויות: a_2 או a_{n-1} שנמצאים במקומות זוגיים.

כאשר B בוחר ב-a_2 ל-A ניתן לבחור ב- a_3, כאשר B בוחר ב-a_n ל-A ניתן לבחור ב- a_{n-1}. כלומר ל-A יש אפשרות שוב לבחור במספר שנמצא במקום אי-זוגי ונותן סכום גדול יותר. יחד עם זאת B יכול לבחור רק במספרים הנמצאים במקומות זוגיים ונותנים סכום קטן יותר.

<u>הנחת אינדוקציה:</u> נניח כי בשלב ש- A בוחר במספר בסדרה נשארו מספרים $a_i, a_{i+1}, \dots, a_{j-1}, a_j$ כאשר i זוגי, ו- j אי-זוגי. מקרה שבו i אי-זוגי, ו- j זוגי – סימטרי לחלוטין.

<u>שלב אינדוקציה:</u> ברור ש-A יבחר ב- a_j, ול-B שוב נשארים שני מספרים a_i ו- a_{j-1} שנמצאים במקומות זוגיים הנותנים סכום קטן יותר. בכל בחירה של B ל-A נפתחת אפשרות לבחור במספר (a_{i+1} או a_{j-2}) שנמצא במקום אי-זוגי ונותן סכום גדול יותר.

שאלה: האם האסטרטגיה הזו נותנת ל-A רווח גדול ביותר?

תשובה: נתבונן בדוגמה: 1,3,6,1,3,6. כאן $S_1=1+3+6=10$, $S_2=3+1+6=10$, האסטרטגיה מביא לתיקו.

אבל ברור ש- A יכול לנצח עם רווח גדול יותר:

B	A	step
$a_5=3$	$a_6=6$	1
$a_2=3$	$a_1=1$	2
$a_4=1$	$a_3=6$	3
7	**13**	summa

<u>שיפור של אסטרטגיה</u>. **אסטרטגיה אדפטיבית** (adaptive strategy). אסטרטגיה אדפטיבית – היא אסטרטגיה כזו אשר נקבעת בתהליך של פתרון הבעיה

אסטרטגיה כזו, אשר נקבעת בתהליך של פתרון הבעיה, על בסיס הצטברות של מידע חדש על התוצאות האפשריות של פתרונות שונים.

כדי לשפר את האסטרטגיה נחשב את הסכומים של מספרים הנמצאים במקומות זוגיים ואי-זוגיים בכל שלב המשחק:

דוגמה: שוב נתבונן בדוגמה: 1,3,6,1,3,6.

שלב 1: A בוחר ב- $a_6=6$, B בוחר ב-$a_5=3$, הסכומים החדשים הם: $S'_1 = a_1 + a_3 = 1 + 6 = 7$, $S'_2 = a_2 + a_4 = 3 + 1 = 4$.

שלב 2: ל-A כדאי לקחת מספר ממקום אי-זוגי, כלומר לבחור ב- $a_1 = 1$, אז B בוחר ב- $a_2=3$.

שלב 3: A: $a_3=6$, B: $a_4=1$. הסכום הסופי של A: $6 + 1 + 6 = 13$, B: $3 + 3 + 1 = 7$.

סיכום: A מנצח עם רווח מקסימאלי: $13 – 7 = 6$.

הערה חשובה: שימוש באסטרטגיה אדפטיבית לא דורשת חישוב חדש של הסכומים: מספיק להחסיר מסחום קודם את המספר החדש שנבחר ע"י השחקן. לדוגמה: $S'_1 = S_1 - a_5 = 10 - 3 = 7$

שאלה: האם האסטרטגיה האדפטיבית תמיד נותנת רווח מקסימאלי?
תשובה: נתבונן בדוגמה: 5, 4, 1, 5, 6, 4. כאן $S_1 = 13, S_2=12$.

אסטרטגיה אדפטיבית:

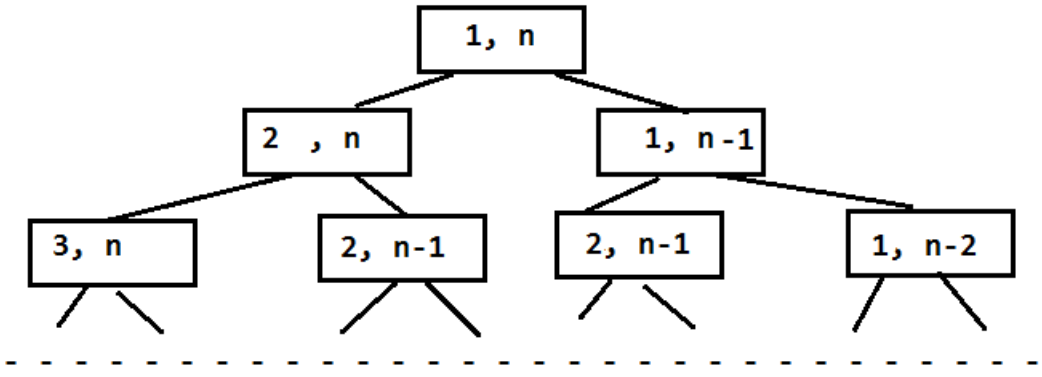
step	1 (odd)	2 (odd)	3	summa
A	$a_6 = 4$	$a_4 = 5$	$a_2 = 4$	13
B	$a_5 = 6$	$a_1 = 5$	$a_3 = 1$	12

אסטרטגיה אופטימאלית:

step	1	2	3	summa
A	$a_1 = 5$	$a_6 = 4$	$a_4 = 5$	14
B	$a_2 = 4$	$a_5 = 6$	$a_3 = 1$	11

<u>קיימת אסטרטגיה ששימוש בה תמיד מביא A לניצחון עם הרווח המקסימאלי</u>.
מניחים כי שני השחקנים חזקים ובכל שלב כל אחד מהם בוחר בפתרון הטוב ביותר.
נעבור על אסטרטגיה של **חיפוש שלם**.

למטרה לבנות אלגוריתם שעובר על כל המצבים האפשריים בונים עץ של סדרה $a_1, a_2, \ldots, a_n$:



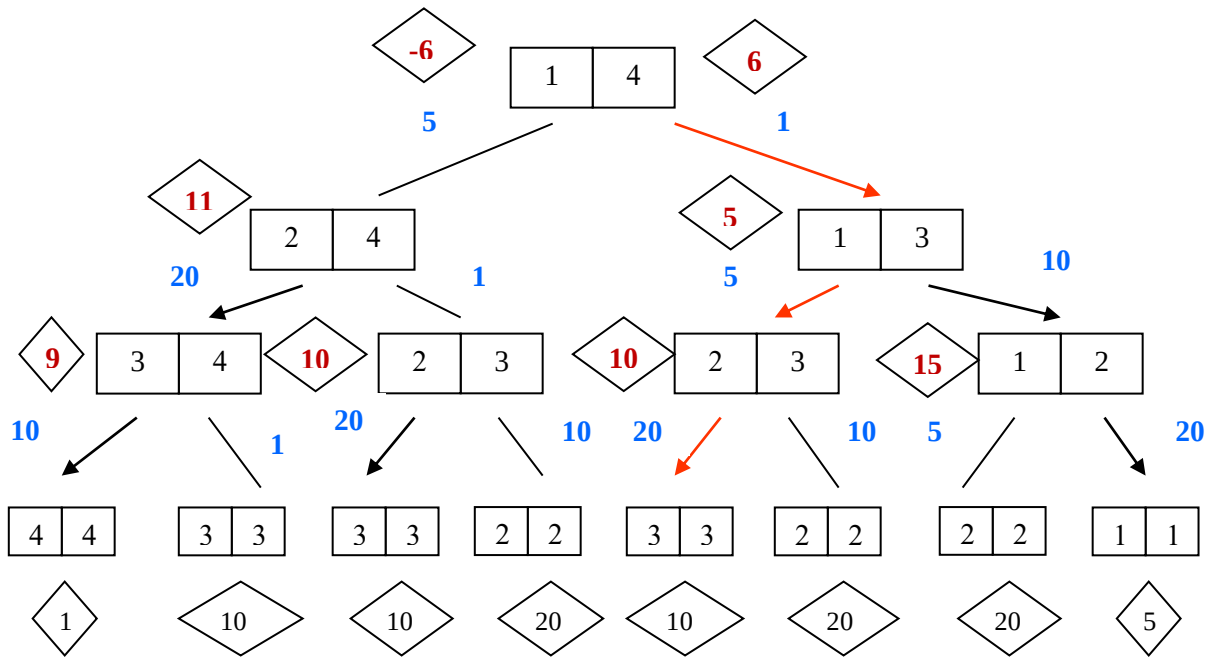
קבלנו עץ בינארי שלם, שגובו הוא **n**, דהיינו מספר צמתים בעץ הוא 2^n-1. למשל, בסדרה של עשרה מספרים מספר אפשרויות (צמתים) הוא 2^n-1 ≅ 1000. כדי לחשב את הרווח המקסימאלי לאחר בניית עץ הולכים מלמטה למלה. כותבים רק את הרווח של A.

דוגמה:

נתון מערך של 4 מספרים:

מספר סידורי:	4	3	2	1
מערך :	1	10	20	5

נבנה עץ של כל המסלולים האפשריים:



מסלול אופטימאלי

9 סכום של ראשון

10 מספר שגוררים ממערך

העץ הוא עץ בינארי שלם, לכן ניתן לממש אותו בעזרת מערך (כמו ערמה):

1,4	2,4	1,3	3,4	2,3	2,3	1,2	4,4	3,3	3,3	2,2	3,3	2,2	2,2	1,1

```
/** function parent returns the parent of vertex i */
      parent(int i){return i/2}
/** function leftChild returns the left child of vertex p */
      leftChild(int p){return 2*p}
/** function rightChild returns the right child of vertex p */
      rightChild(int p){return 2*p+1}
```