

ממ"ן 13 - פתרון שאלה 1

```
HEAP-INSERT(A, key)
1   A[0] ← ∞
2   heap-size[A] ← heap-size[A] + 1
3   i ← heap-size[A]
4   while A[PARENT(i)] < key
5       do A[i] ← A[PARENT(i)]
6       i ← PARENT(i)
7   A[i] ← key
```

השגרה תמיד תעבוד נכון : אם מגיעים לשורש ($i = 1$), אז $\text{PARENT}(i) = 0$ והתנאי $A[0] < \text{key}$ לא מתקיים, כך שהלולאה תסתיים.

ממ"ן 13 - פתרון שאלה 2

א'

```
HEAP-INCREASE-KEY (A, i, k)
1   A[i] ← max(A[i], k)
2   key ← A[i]
3   while i > 1 and A[PARENT(i)] < key
4       do A[i] ← A[PARENT(i)]
5       i ← PARENT(i)
6   A[i] ← key
```

במקרה הגרוע i עולה עד השורש, ולכן זמן הריצה של השגרה הוא $O(\lg n)$.

ב'

```
HEAP-DECREASE-KEY (A, i, k)
1   A[i] ← min(A[i], k)
2   HEAPIFY(A, i)
```

ג'

```
HEAP-DELETE (A, i)
1   k ← A[heap-size[A]]
2   heap-size[A] ← heap-size[A] - 1
3   if k < A[i]
4       then A[i] ← k
5       HEAPIFY(A, i)
6   else while i > 1 and A[PARENT(i)] < k
7       do A[i] ← A[PARENT(i)]
8       i ← PARENT(i)
9   A[i] ← key
```

ד'

```
HEAP-DELETE1 (A, i)
1   k ← A[1] + 1
2   HEAP-INCREASE-KEY(A, i, k)
3   HEAP-EXTRACT-MAX(A)
```