Fundamental Algorithms, Problem Set 1 Solutions

1. Let A is a max-heap with heapsize fifty million, being used as a priority queue. Suppose `HEAP-INCREASE-KEY(A,300,key)` is called. What is the maximum number of exchanges that can take place. What is the minimal number of exchanges that can take place.
   `Solution:`The minimal number is zero, if the new key has smaller value than the key of the parent of 300. The maximal number is achieved when the new key is the largest key in the heap, and so the exchanges go up to the root. So there are exchanges with 150, then 75, then 37, then 18, then 9, then 4, then 2, then 1 for a total of eight exchanges. (In general the position $A[i]$ is on row j (counting the root as row zero) where $2^j \le i < 2^{j+1}$. Reversing, $j = \lfloor \lg i \rfloor$. The maximum number of exchanges would then be j.

2. When A is a array with length fifty million and `MAX-HEAPIFY(A,300)` is called. What is the maximum number of exchanges that can take place. What is the minimum number of exchanges that can take place.
   `Solution:`With `MAX-HEAPIFY` we work *down* to the leaves. The minimum is zero, if the key at 300 is bigger than the keys of both its children. The maximum would be if we have to work all the way down the the leaves. We could exchange with $600, 1200, 2400, 4800, 9600, 19200, 38400$ then $76800, 153600, 307200, 614400$ then (in rough millions) $1.3, 2.6$,5.2, $10.4, 20.8, 41.6$. So the total would be seventeen exchanges. (One might also exchange with the right children, but as we are examining worst case we consider the left children as they have smaller indexes.) In general, if the array has length n then starting at i there is a descendent j generations down if and only if $i2^j \le n$, or, equivalently, $j \le \lg(n/i)$. So the maximal number would be $\lfloor \lg(n/i) \rfloor$.

3. Consider a min-heap H with length 1023. [1] Assume the elements of the array are distinct. Let x be the third smallest element in the array. What are the possible positions for x.
   `Solution:`It could be in any position in the first two rows, that is, from two to seven.
   Let $y = H[700]$. Can y be the largest element in the array?
   `Solution:`Sure, any leaf could be the largest in a min-heap.
   Can y be the smallest element in the array?

[1]Did you recognize 1023 as a special number? Its one less than $1024 = 2^{10}$. The binary tree with that many nodes just fills out a row!

`Solution:`No way, the smallest in a min-heap must be at the root.
(hard!) Give all i for which it is possible that y is the i-th smallest element of the array.
`Solution:`Its ancestors $(350, 175, 87, 43, 21, 10, 5, 2, 1)$ must be smaller so it can be at best tenth smallest. Its descendents must be bigger but it doesn't have any descendents. A full proof is a bit complicated (did you find it?!) but these are the only conditions and it could be the i-th smallest for any $10 \le i \le 1023$.

4. Using the figures in the text as a model, illustrate the operation of `BUILD-MAX-HEAP` on the array $A = (5, 3, 17, 10, 84, 19, 6, 22, 9)$

5. The operation `HEAP-DELETE(A,t)` deletes the item in node t from heap A. Give an implementation of `HEAP-DELETE` that runs in $O(\lg n)$ time for an n-element max-heap.
`Solution:`First a simple case: If t is $A.heapsize$ then simply decrement $A.heapsize$. ELSE first reset $A[t] \leftarrow A[A.heapsize]$ and decrement $A.heapsize$. Now you have the right elements but $A[t]$ may be in the wrong places. First check if $A[t] > A[parent[t]]$ (Ignore this if t is the root.) If so, we have a WHILE loop, exchanging $A[t]$ and $A[parent[t]]$ and then resetting $t \leftarrow parent[t]$ while t is not the root and $A[t] > A[parent[t]]$. ELSE (that is, if we did not have $A[t] > A[parent[t]]$, then the only (possible) problem is that $A[t]$ is too small. So we apply $MAX - HEAPIFY[A, t]$.

6. Let A be an array of length 127 in which the values are distinct and in increasing order. In the procedure `BUILD-MAX-HEAP(A)` precisely how many times will two elements of the array be exchanged?
`Solution:`BUILD-MAX-HEAP(A) starts from I = LENGTH(A)/2 DOWN to 1, every I will do Max-Heapify.
For $32 \le I \le 63$,there should be one exchange.
For $16 \le I \le 31$, there should be 2 exchanges.
For $8 \le I \le 17$, there should be 3 exchanges.
For $I = 4, 5, 6, 7$, there should be 4 exchanges.
For $I = 2, 3$ there should be 5 exchanges.
The root goes down to the bottom, 6 exchanges.
Total: $32 \cdot 1 + 16 \cdot 2 + 8 \cdot 3 + 4 \cdot 4 + 2 \cdot 5 + 1 \cdot 6 = 120$
Now suppose the values are distinct and in decreasing order. Again, in the procedure `BUILD-MAX-HEAP(A)` precisely how many times will two elements of the array be exchanged?
`Solution:`Never! Each element will be placed originally in precisely

its correct final spot.