$$\sum_{i=0}^{k} 2^i (k-i) = \sum_{i=0}^{k} 2^i \cdot k - \sum_{i=0}^{k} i \cdot 2^i$$

$$S = 1\cdot 2 + 2\cdot 4 + 3\cdot 8 + 4\cdot 16 + \cdots + k\cdot 2^k$$

$$2S = \quad 1\cdot 4 + 2\cdot 8 + 3\cdot 16 + \cdots (k-1)\cdot 2^k + (k\cdot 2^{k+1})$$

$$S - 2S = 1\cdot 2 + 1\cdot 4 + 1\cdot 8 + \cdots + 1\cdot 2^k - k\cdot 2^{k+1}$$

$$-S = (2 + 4 + 8 + \cdots + 2^k) - k\cdot 2^{k+1}$$

$$-S = \frac{2^{k+1} - 2}{2-1} - k\cdot 2^{k+1}$$

$$-S = 2^{k+1} - 2 - k\cdot 2^{k+1} = -2 - (k-1)\cdot 2^{k+1}$$

$$S = 2 + (k-1)\cdot 2^{k+1}$$

$$k\cdot(2^{k+1} - 1) -$$

$$k\cdot(2^{k+1} - 1) - \left[2 + (k-1)\cdot 2^{k+1}\right]$$

$$k\cdot 2^{k+1} - k - 2 - (k-1)\cdot 2^{k+1}$$

$$\underbrace{k\cdot 2^{k+1} - (k-1)\cdot 2^{k+1}} - k - 2$$

$$2^{k+1}(k - (k-1)) - (k+2)$$

$$2^{k+1} - k - 2 = 2\cdot 2^k - k - 2$$

$$(k = \log_2 n)$$

$$2 \cdot 2^k - k - 2 = 2 \cdot n - \log_2 n - 2 = \Theta(n)$$!

$(k = \log_2 n)$

So we can build a max-heap in linear time!!

Heapsort:

In a max-heap, the largest element is always at the top.

Heap Sort

for $i \leftarrow n$ downto 2

do exchange $A[i] \longleftrightarrow A[1]$

heapsize $\leftarrow$ heapsize -1

max-heapify$(A, 1)$