On the number of heaps

Hsien-Kuei Hwang
LIX, École polytechnique
February 8, 1993

[summary by Hsien-Kuei Hwang]

Abstract

The main interest in this talk is the asymptotic behaviour of the number of heaps of size \( n \) as \( n \to \infty \). For special sequences of \( n \), like \( \{2^k\}_k \) or \( \{2^k - 1\}_k \), the result is easily obtained by resolving linear recurrences of first order. In order to obtain a general asymptotic formula, we need to introduce some oscillating digital sums (depending on the digits of the binary representation of \( n \)) whose behaviours can only be grasped by their summatory functions which are more manageable.

1. Heap Recurrences

A (max-)heap is an array with elements \( a_j, 1 \leq j \leq n \), satisfying the path-monotone property: \( a_j \leq a(j/2) \), \( j = 2, 3, \ldots, n \). It can be viewed as a binary tree where the value of each element is not smaller than that of its children. A characteristic property of a heap, when viewed as a binary tree, is that at least one of the two sub-trees of the root node is complete (i.e., it contains \( 2^k - 1 \) elements for some non-negative integer \( k \)). And this property recursively applies to each node. Given a heap \( \mathcal{H}_n \) of size \( n \) and an additive cost function \( \varphi \) on heaps, we have the relation

\[
\varphi[\mathcal{H}_n] = \tau[\mathcal{H}_n] + \varphi[\mathcal{H}_L] + \varphi[\mathcal{H}_R],
\]

for some cost function \( \tau \), where \( \mathcal{H}_L \) and \( \mathcal{H}_R \) denote the left and right sub-heaps of the root node of \( \mathcal{H}_n \) with sizes \( L \) and \( R \), respectively. Since at least one of \( \mathcal{H}_L \) or \( \mathcal{H}_R \) is complete, the relation (1) can be written into a more precise form as follows. For \( k \geq 0 \) and \( \{t_n\}_{n \geq 1} \) a given non-negative sequence,

\[
\begin{cases}
  f_{2k} = t_{2k} + \left\{ 
  f_{2^{k-1} - 1} + f_{2^{k-1} + j}, & \text{if } 0 \leq j < 2^{k-1}, \\
  f_{2^{k-1}} + f_j, & \text{if } 2^{k-1} \leq j < 2^k,
  \end{cases}
\]

(2)

which we call the additive heap recurrence [3]. The associated generating functions are not very suggestive for further investigations.

\[
f(z) = \sum_{n \geq 1} t_n z^n + \frac{1}{1-z} \sum_{k \geq 1} f_{2^k} \left( z^{2^k-1} - z^{2^k} \right) + \sum_{k \geq 1} \left( z^{2^k} + z^{2^k-1} \right) \sum_{2^k \leq j < 2^{k+1}} f_j z^j,
\]

where \( f(z) = \sum_{n \geq 1} f_n z^n \).

Let \( h_n \) denote the total number of ways to rearrange the integers \( \{1, 2, \ldots, n\} \) into a heap. Then it is obvious that \( h_n \) satisfies the multiplicative heap recurrence:

\[
h_{2k+1} = \begin{cases}
  (2^k + j - 1)^{h_{2^k-1} h_{2^k-1} + j}, & \text{if } 0 \leq j < 2^{k-1}, \\
  (2^k - 1)^{h_{2^k-1} h_j}, & \text{if } 2^{k-1} \leq j < 2^k.
\end{cases}
\]

135
IV  Analysis of Algorithms and Data Structures

The sequence
\[
\{h_n\}_{n \geq 2} = 1, 2, 3, 8, 20, 80, 210, 896, 3360, 19200, 79200, 506880, 2745600, \\
21964800, 108108000, 820019200, 5227622400, 48881664000... 
\]

is not in Sloane’s book. Let \( f_n = \log(n!/h_n) \), then \( f_n \) satisfies the additive heap recurrence. We require then to find the general solution of (2).

Let us first fix some notations.
- \( n \) is a positive integer, and \( n = (b_L b_{L-1} \ldots b_1)_{b_2} \), where \( \lfloor \log_2 n \rfloor \) and \( b_L = 1 \).
- \( n_j = (b_{j-1} \ldots b_0)_{b_2} \) for \( j = 1 \ldots \); \( n_0 = 1 \).
- \( \nu(n) \) denotes the number of 1-digits in the binary representation of \( n \).

Before solving (2), we note that there is another very similar type of recurrences [2]

\[
\phi_{2^k+j} = \tau_{2^k+j} + \left\{ \begin{array}{ll}
\phi_{2^k-1} + \phi_{2^k-1+j}, & \text{if } 0 \leq j < 2^{k-1}; \\
\phi_{2^k} + \phi_j, & \text{if } 2^{k-1} \leq j < 2^k, 
\end{array} \right.
\]

which occurs as the solution of the following equation

\[
\phi_n = \tau_n + \min_{1 \leq j \leq \lfloor \log_2 n \rfloor} \phi_{j + \nu(n - j)},
\]

when the sequence \( \{\tau_n\}_{n \geq 0} \) is strictly concave, namely \( \tau_{n+2} - 2\tau_{n+1} + \tau_n < 0 \) for all \( n \geq 0 \).

Recall that the backward difference is defined by \( \nabla f_n = f_n - f_{n-1} \). Let \( \varphi_n = \nabla f_n \), and \( \tau_n = \nabla t_n \), then we obtain a slightly different recurrence

\[
\varphi_{2^k+j} = \tau_{2^k+j} + \left\{ \begin{array}{ll}
\varphi_{2^k-1} + \varphi_{2^k-1+j}, & \text{if } 0 \leq j < 2^{k-1}; \\
\varphi_{2^k} + \varphi_j, & \text{if } 2^{k-1} \leq j < 2^k, 
\end{array} \right.
\]

together with \( \varphi_0 = 0 \). Equivalently, this recurrence can be re-written as \( \varphi_n = \varphi_{n_L} = \tau_n + \varphi_{n_L-1} = \sum_{0 \leq j \leq L} \tau_{n_j} \).

2. Explicit Formula

To solve the heap recurrence explicitly, we first observe that when \( n = 2^{m+1} - 1 \), we have a linear recurrence: \( f_{2m+1} = f_{2m+1-1} + 2 f_{2m} \), which can be solved easily by iteration. From this, we can find the solution for the sequences \( \{2^m\}, \{2^m + 2^{m+1} - 1\}, \ldots \). But this process does not lead readily to a general solution. Hence, we begin with another way.

**Lemma 1.** For \( n \geq 1 \), we have, for the solution of (2),

\[
f_n = \sum_{1 \leq j \leq L} \left( \left\lfloor \frac{n}{2^j} \right\rfloor - \left\lfloor \frac{n}{2^j} \right\rfloor - 1 \right) t_{2^j-1} + \sum_{0 \leq j \leq L} t_{n_j}.
\]

The two sums correspond, respectively, to the contribution of complete sub-heaps and non-complete sub-heaps.

Similarly, the solution for the recurrence (3) is expressed by \( \phi_0 = 0 \)

\[
\phi_n = \sum_{0 \leq j \leq L} \left( \left\lfloor \frac{n}{2^j} \right\rfloor - \left\lfloor \frac{n}{2^j} \right\rfloor - 1 \right) \tau_{2^j} + \sum_{0 \leq j \leq L} \tau_{n_j}.
\]

An immediate consequence of Lemma 1 is the following

**Lemma 2.** Let \( t_n > 0 \) and \( t_n = O(n^{1-c}) \) for fixed \( c > 0 \), then the solution \( f_n \) of (1) satisfies \( f_n \sim cn \), as \( n \) tends to infinity, for some constant \( c \). Moreover, the constant \( c \) is given by

\[
c = \sum_{j \geq 1} \frac{t_{2^j-1}}{2^j}.
\]

\footnote{The series \( \sum_{j \geq 1} \frac{t_{2^j-1}}{2^j} \) is easily seen to be convergent.}
This result says that without loss of generality, we can, under the hypotheses of Lemma 2, consider only the special sequence \( \{2^m - 1\}_m \), as far as the first asymptotic term is concerned.

For recurrence (3), constant \( c \) is modified to be \( c = \sum_{j \geq 0} \tau_{2^j} / 2^j \), under the same conditions.

3. The Number of Heaps

Let \( f_n = \log(n! / h_n) \), then \( f_n \) satisfies (2) with \( t_n = \log n \). Lemma 2 gives the first-order estimate of \( f_n \)

\[
f_n \sim n \sum_{j \geq 1} \frac{\log(2^j - 1)}{2^j} = n \left( 2 \log 2 + \sum_{j \geq 1} \frac{1}{2^j} \log(1 - 1/2^j) \right) = 0.945755... n.
\]

Let \( \alpha = 2 \log 2 + \sum_{j \geq 1} 2^{-j} \log(1 - 2^{-j}) \) be the coefficient. Using Lemma 1, we obtain the main result of this talk.

**Theorem 1.**

\[
h_n \sim 2Q \sqrt{2\pi} P(\log_2 n) R(n) n^{n+\frac{1}{2}} e^{-\alpha n - n} \quad (n \to \infty),
\]

where \( Q = \prod_{j \geq 1} (1 - 2^{-j}) = 0.288788... \),

\[
P(u) = 2^{2(u) - (u)} \prod_{0 \leq j \leq u} \frac{2^{(u-j)}}{1 + [2^{u-j}]},
\]

and

\[
R(n) = \prod_{j \geq 1} \left( \frac{1 - 2^{-j-1}}{1 - 2^{-j}} \right)^{\{n/2^j\}}.
\]

The two functions \( P \) and \( R \) are oscillating in nature. We can prove that, for all \( n \geq 1 \),

\[
1 \leq R(n) \leq \exp \left( - \sum_{j \geq 1} 2^{-j} \log(1 - 2^{-j}) \right) = 1.553544...
\]

and

\[
0 < 2^{-\{\log_2 n\} + c_0 \nu(n)} < P(\log_2 n) \leq 2,
\]

where \( c_0 = 1 - c_1 / \log 2 = -0.253522... \) with \( c_1 = \sum_{j \geq 1} \log(1 + 2^{-j}) = 0.868876... \).

To further investigate the properties of the two functions \( R \) and \( P \), we observe that \( R \) is bounded for all \( n \). For \( P \), let \( p(n) = \log P(\log_2 n) \), then

\[
p(n) = \nu(n) - \{\log_2 n\} - \sum_{0 \leq j \leq \log_2 n} \log_2(1 + \{n/2^j\}),
\]

so that \( p \) oscillates between \( O(\log n) \) and \( O(1) \). Since the first two terms on the right-hand side are “known”, only the last sum needs special treatments. Set \( \pi(n) = \sum_{0 \leq j \leq \log_2 n} \log(1 + \{n/2^j\}) \). Then, for \( x \) not an integer, we have the convergent Fourier series

\[
\log(1 + \{x\}) = 2 \log 2 - 1 + \sum_{k \neq 0} \frac{e^{2k\pi i x}}{2k\pi i} (\text{Ei}(-4k\pi i) - \text{Ei}(-2k\pi i) - \log 2).
\]

For \( x \) an integer, the series converges to \( \frac{1}{2} \log 2 \). \( \text{Ei}(z) \) is the exponential integral. Now summing all such series for \( j = 1, 2, \ldots, L \), we obtain

\[
\pi(n) = (2 \log 2 - 1) - \frac{\log 2}{2} v_2(n) + \sum_{k \neq 0} \frac{\text{Ei}(-4k\pi i) - \text{Ei}(-2k\pi i) - \log 2}{2k\pi i} \sum_{1 \leq j \leq L} e^{2k\pi i j / 2^j},
\]

137
IV Analysis of Algorithms and Data Structures

which is a mere translation of $\pi(n)$ into trigonometric sums. Here $v_2(n)$ denotes the exponent of 2 in the prime decomposition of $n$. Yet the formula still says something about the average order of $\pi(n)$:

$$\frac{1}{n} \sum_{1 \leq m \leq n} \pi(m) = (2 \log 2 - 1) \log_2 n + O(1) \quad (n \to \infty),$$

which can be obtained by the following “Ergodic-type” result.

\textbf{Theorem 3.} For any real continuous function $\varphi(x)$ on $[0,1]$, define $\phi(m) = \sum_{0 \leq j \leq \log_2 m} \varphi(\{m/2^j\})$. We have the asymptotic formula

$$\frac{1}{n} \sum_{1 \leq m \leq n} \phi(m) = \left( \int_0^1 \varphi(x) \, x \right) \log_2 n + O(1) \quad (n \to \infty).$$

In words, the lemma says that the average order of the function $\phi(m)$ is asymptotically equal to $\log_2 n$ times the mean value of the function $\varphi$ on $[0,1]$.

. The Cost of Constructing Heaps

Given a random permutation $\pi_n$ of size $n$, let $\xi_n$ denote the number of exchanges used to construct a heap from $\pi_n$ using Floyd’s algorithm. Then $E\xi_n$ satisfies the heap recurrence with $t_n = n^{-1} \sum_{1 \leq j \leq n} \log_2 j = 1 + (\log 2)/n - 2L^{1+1}/n$. Applying Lemma 1, we get the following refined result of Sprugnoli [3], who considered only special sequences of $n$.

\textbf{Theorem 2.} The expected number of exchanges $E\xi_n$ used in Floyd’s heap construction algorithm satisfies

$$E\xi_n = c_2 n - \left( \log_2 n \right) - \nu(n) + 2 \varpi_1(n) + \varpi_2(n) + O\left( \frac{\log n}{n} \right) \quad (n \to \infty),$$

where $c_2 = -2 + \sum_{j \geq 1} j(2^j - 1)^{-1} = 0.744033...$, $\varpi_1(n)$ oscillates between $O(\log n)$ and $O(1)$,

$$\varpi_1(n) = \sum_{0 \leq j \leq L} \frac{n/2^j}{1 + \{n/2^j\}},$$

and $\varpi_2(n) = O(1)$ is given by

$$\varpi_2(n) = 1 - \sum_{j \geq 1} \frac{j}{2^j - 1} + \sum_{j \geq 1} \frac{j^2}{2(1 + \{n/2^j\})} + \sum_{j \geq 1} \left\{ \frac{n}{2^j} \right\} \frac{j^2 - 2^j + 1}{(2^j - 1)(2^j + 1 - 1)}.$$

In particular, we have the inequalities $\frac{1}{2} (\nu(n) - n/2^L) \leq \varpi_1(n) \leq c_3 \nu(n)$ for all $n$, so that

$$c_2 n - \nu(n) + O(1) \leq \xi_n \leq c_2 n - (2c_3 - 1) \nu(n) + O(1),$$

for all $n$, where $c_3 = \sum_{j \geq 1} (2^j + 1)^{-1} = 0.764499...$ and $2c_3 - 1 = 0.528999...$

By Lemma 3, the average order of the arithmetic function $\varpi_1(n)$ is $(1 - \log 2) \log_2 n + O(1)$.

For the variance, we take

$$t_n = \frac{1}{n} \sum_{1 \leq j \leq n} \left( \log_2 j \right)^2 - \left( \frac{1}{n} \sum_{1 \leq j \leq n} \log_2 j \right)^2$$

$$= 6 \frac{2 L}{n} - \frac{2}{n} - 4 \frac{1}{n} - 4 \frac{1}{n^2} - 4 \frac{1}{n^2} + \frac{2 L + 3}{n^2} - \frac{2}{n^2} + \frac{2 L + 2}{n^2} - \frac{4 L + 1}{n^2}.$$
Theorem 3. The variance of the number of exchanges satisfies the asymptotic expression
\[
\text{Var}(\xi_n) = c_4 n + \varpi_3(n) + \varpi_4(n) + O \left( \frac{\log^2 n}{n} \right) \quad (n \to \infty),
\]
where \( c_4 = 2 - \sum_{j \geq 1} j^2 (2^j - 1)^2 = 0.261217 \ldots \), \( \varpi_3(n) \) oscillates between \( O(\log n) \) and \( O(1) \):
\[
\varpi_3(n) = -2 \sum_{0 \leq j \leq \log_2 n} \frac{\{n/2^j\}}{1 + \{n/2^j\}} + 4 \sum_{0 \leq j \leq \log_2 n} \frac{\{n/2^j\}}{(1 + \{n/2^j\})^2},
\]
and \( \varpi_4(n) = O(1) \):
\[
\varpi_4(n) = \sum_{j \geq 1} j^{2j} \left( \frac{n}{2^j} \right)^2 + \sum_{j \geq 1} j^{2j} \left( \frac{n}{2^j} \right)^2 \left( \frac{2^j (j^2 + 4j + 2) - 4^{j+1} (2j + 1) - 2 \cdots 8^j (j^2 - 2j - 1)}{(2j - 1)^2 (2j + 1 - 1)^2} \right).
\]

The average order of \( \varpi_3(n) \) is \( (6 \log 2 - 4) \log_2 n + O(1) \).

Finally, from the probability generating function of \( \xi_n \) derived in [1], it is not hard to show that the distribution of \( \xi_n \) is asymptotically Gaussian.

Theorem 4. We have
\[
\Pr \left\{ \frac{\xi_n - c_2 n}{\sqrt{c_4 n}} < x \right\} = \frac{1}{\sqrt{2\pi}} \int_{-\infty}^{x} e^{-\frac{1}{2} t^2} \ dt + O \left( \frac{\log n}{\sqrt{n}} \right) \quad (n \to \infty),
\]
uniformly with respect to \( x \).

Bibliography

