JOURNAL OF ALGORITHMS **12**, 126-153 (1991)

Average Case Analysis of Heap Building by Repeated Insertion

RYAN HAYWARD*

Computer Science Dept., Rutgers University, New Brunswick, N.J. 08903 U.S.A.

AND

COLIN MCDIARMID

Department of Statistics, Oxford University, England

Received November 22, 1988; revised January 2, 1990

We show that the average number of swaps required to construct a heap on n keys by Williams' method of repeated insertion is $(\omega + o(1))n$, where the constant ω is about 1.3. Further, with high probability the number of swaps is close to this quantity. These results build on earlier work of Bollobás and Simon, and of Frieze.
©1991 Academic Press, Inc.

1 Introduction

An array $A[1], \ldots, A[n]$ is a (min) *heap* if $A[\lfloor i/2 \rfloor] \leq A[i]$, for $i = 2, \ldots, n$. The heap is a much used and much studied data structure (for example, see Knuth [K]).

Williams' method for constructing a heap involves repeatedly inserting a key at the bottom of the heap and 'bubbling' it up. This method requires $\Theta(n \log n)$ time in the worst case. Let W_n denote the number of 'swaps' (or 'promotions') performed in the building of a heap with n keys (see Appendix 1 for a precise definition).

We make the usual assumption that each initial permutation of the keys is equally likely. It is shown by Bollobás and Simon [BS] and Frieze [F] that for perfect heaps (that is, for n equal to $2^k - 1$) the expected number $E[W_n]$

* Research supported in part by an Alexander von Humboldt-Stiftung fellowship.

of swaps satisfies

$$(0.75 + o(1))n \leq E[W_n] \leq (1 + \phi)n \tag{1.1}$$

where $\phi = \sum_{j \geq 1} 1/(1 + 2^j) \approx 0.7645$. It is also shown in [F] that for any $\varepsilon > 0$ there is a constant $\delta > 0$ such that

$$\text{Prob}\{W_n/n > (1 + \phi + \varepsilon)n\} < \exp(-\delta n^{1/10}). \tag{1.2}$$

At the recent Bellairs Research Workshop on Combinatorics (1988) Frieze asked whether better bounds could be found for $E[W_n]$, and a slight improvement in the lower bound was obtained by Devroye [De]. We show the following result.

Theorem 1.3

(a) There exists a constant ω such that $E[W_n]/n \to \omega$ as $n \to \infty$.
(b) $1.2778\ldots < \omega < 1.2994\ldots$.
(c) For any $\varepsilon > 0$, $\text{Prob}\{|W_n/n - \omega| > \varepsilon\} = o(\exp(-n/\log^4 n))$.

Parts (a) and (b) strengthen (1.1), and part (c) strengthens (1.2). Extrapolation of numerical results suggests that $\omega \approx 1.283$.

In order to compare Williams' repeated insertion method with other methods of heap construction, let us restate parts (a) and (b) in terms of the number C_n of key comparisons. Call a key a 'new minimum' if it is smaller than all previous keys. Note that $C_n = W_n + n - M_n$, where M_n is the number of 'new minima', and $E[M_n] = 1 + 1/2 + \ldots + 1/n \approx \ln(n)$. (Throughout the paper we use $\ln(n)$ for the log base e and $\lg(n)$ for the log base 2.) Hence we have

$$E[C_n]/n \to 1 + \omega, \text{ where } 2.2778 < 1 + \omega < 2.2994. \tag{1.4}$$

Floyd's method of heap building (see Floyd [F] or Knuth [K]) involves repeatedly merging small heaps to form bigger heaps. This method requires $(2+o(1))n$ comparisons in the worst case to build a heap with n keys, which is less than Williams' method takes on average. The average-case behaviour of Floyd's method, and variants, has been well studied. Knuth [K] (see also Doberkat [Do]) shows that the expected number of comparisons for the basic method is about $1.88n$. Carlsson [C] and independently McDiarmid and Reed [MR] show that by changing the 'trickledown' method this expected number drops to about $1.65n$. Further, it is shown in [MR] that if all comparisons are remembered, then the two numbers drop to about $1.79n$ and $1.52n$ respectively, and for each variant there is strong concentration around

the mean. For further discussion on heaps, and for the heap-building method with the best known worst-case number of comparisons (namely $1.625n$), see [GM].

This paper is organized as follows. In the next section we develop some qualitative results useful for investigating the expected number of swaps $E[W_n]$ and show that the Williams' heap building constant ω exists (part (a) of the theorem). The key observation of this section, Observation 2.1, is the cornerstone of our analysis. In section 3 we present a polynomial time algorithm to compute $E[W_n]$ and related quantities, and in section 4 we use various computed quantities to establish bounds on ω (part (b) of the theorem). Section 5 is devoted to proving part (c) of the theorem. Section 6 contains a brief discussion concerning repeated insertion into equiprobable heaps, as a possible approximate analysis of Williams' method (which does not yield equiprobable heaps). Finally, in section 7 we present a few concluding remarks.

2 Qualitative results for expected times

In this section we show that the Williams' heap building constant ω does indeed exist (part (a) of Theorem 1.3), and give some results which will be used later to establish bounds on ω. We shall build on the work of Bollobás and Simon [BS] and Frieze [F], although our treatment here is self-contained.

We wish to investigate the behaviour of Williams' algorithm when it builds a heap on n random keys $A[1], \ldots, A[n]$ where each of the $n!$ linear orders on these n keys is equally likely. There are (at least) two natural ways to model the input distribution for the purposes of average case analysis. One is to assume that the set of keys is $\{1, \ldots, n\}$ and that each of the $n!$ possible input permutations is equally likely. Another is to assume that the keys are n independent random variables, each uniformly distributed on the interval [0,1]. Since the only operations performed on keys are comparisons, both models are valid. The former model is used by Bollobás and Simon [BS]; Frieze [F] found that some arguments are shorter if the latter model is used. This is the case here as well, and so we use the 'independent uniform' model.

Thus, following [F] we suppose that our array $A[1], \ldots, A[n]$ is the initial segment of an infinite array, and assume that the initial contents form a sequence of independent random variables each uniformly distributed on [0,1]. The array A corresponds to a binary tree T where node 1 is the root, node i has children $2i$ and $2i+1$, and the *contents* of node i is $A[i]$. The

level L_k consists of all nodes t with $2^k \leq t < 2^{k+1}$.

The following is our key observation. Say that arrays $A[1], \ldots, A[n]$ and $B[1], \ldots, B[n]$ have *property* m, where $1 \leq m \leq n$, if

$A[1], \ldots, A[m]$ and $B[1], \ldots, B[m]$ are heaps,

$A[i] \geq B[i]$ for $1 \leq i \leq m$, and

$A[i] = B[i]$ for $m+1 \leq i \leq n$.

Observation 2.1 *Suppose that property m holds, and that key $m+1$ is bubbled up in each array. Then property $(m+1)$ holds, and if a tree link is traversed with array B then this also happens with array A.*

Lemma 2.2 *Let $A[1], \ldots, A[n]$ be an array with corresponding binary tree T, and let T' be a subtree of T. Let $(A[i] : i \in T')$ record the contents of the nodes i in T' as we apply Williams' algorithm to T, and let $(A'[i] : i \in T')$ record these values as we work with T' on its own. Then*

(a) for each node $i \in T'$, the number of swaps when i is bubbled up in T' is at most the number of swaps in T' when i is bubbled up in T;
(b) after each bubble up, $A[i] \geq A'[i]$ for each $i \in T'$.

Proof Bubble up all nodes $i < v$, where v is the root of the subtree T'. Then let $B[i] = 0$ for $i < v$ and $B[i] = A[i]$ for $i \geq v$. Now use Observation (2.1). For each node i in T',

	the number of swaps	when i is bubbled up in T'
$=$	the number of swaps	when i is bubbled up with array B
$\leq$	the number of swaps in T'	when i is bubbled up with array A
$=$	the number of swaps in T'	when i is bubbled up in T.

Also, for each node i in T', $A'[i] = B[i] \leq A[i]$. □

Now we need some notation (rather a lot!).

Our interest is in the number $X(t)$ of swaps when node t is bubbled up. For each node t in level L_k $(k > 0)$ let $IP(t) = \{\lfloor t/2^j \rfloor : j = 1, \ldots, k\}$ be the set of nodes on the insertion path from node t to the root. Thus $X(t)$ is the number of nodes i in $IP(t)$ such that $A[t] < A[i]$ when t is about to be bubbled up.

It is convenient to introduce an upper bound for $X(t)$. Let $Y(t)$ be the number of nodes i in $IP(t)$ such that $A[t] < A[i]$ at the stage when we are about to start bubbling up (the first node on) level L_k. Thus $X(t) \leq Y(t)$.

Both Bollobás and Simon [BS] and Frieze [F] work with $Y(t)$ rather than $X(t)$.

Let $X_k = \sum_{t \in L_k} X(t)$ be the total number of swaps when all the nodes in level L_k are bubbled up, and let $x_k = 2^{-k} E[X_k]$ be the corresponding average expected value. Similarly, let $Y_k = \sum_{t \in L_k} Y(t)$ and $y_k = 2^{-k} E[Y_k]$.

Next consider the values $A[t]$ at the time when we have just bubbled up (the last node on) level L_k. Let $A_{j,k}$ denote the sum of these values over all nodes t in L_j, and let $a_{j,k} = 2^{-j} E[A_{j,k}]$ be their average expected value. Finally, let $B_{j,k} = \sum_{i=0}^{j} A_{i,k}$.

Lemma 2.3

$$\text{(a)} \quad 2^{-k} E[Y_k | A[j], j = 1, \ldots, 2^k - 1] \; = \; \sum_{j=0}^{k-1} 2^{-j} A_{j,k-1} \; ,$$

where $A[j]$ is the contents of node j just before bubbling up the first node on level L_k.

$$\begin{aligned} \text{(b)} \quad y_k \; &= \; \sum_{j=0}^{k-1} a_{j,k-1} \\ &= \; 2^{-(k-1)} E[B_{k-1,k-1}] \quad + \quad \sum_{j=0}^{k-2} 2^{-(j+1)} E[B_{j,k-1}] \; . \end{aligned}$$

Proof (a) With $A[j]$ as above, for each node t in L_k

$$\begin{aligned} &E[Y(t) | A[j], j = 1, \ldots, 2^k - 1] \\ &\quad = \sum_{i \in IP(t)} \text{Prob}\{A[t] < A[i] | A[j], 1 \le j \le 2^k - 1\} \\ &\quad = \sum_{i \in IP(t)} A[i]. \end{aligned}$$

Hence

$$\begin{aligned} &E[Y_k | A[j], j = 1, \ldots, 2^k - 1] \\ &\quad = \sum_{t \in L_k} \sum_{i \in IP(t)} A[i] \\ &\quad = \sum_{j=0}^{k-1} 2^{k-j} A_{j,k-1} \; , \end{aligned}$$

which gives part (a).
(b) Remove the conditioning in part (a) to obtain

$$\begin{aligned} y_k &= \sum_{j=0}^{k-1} 2^{-j} E[A_{j,k-1}] \quad = \quad \sum_{j=0}^{k-1} a_{j,k-1} \\ &= E[B_{0,k-1}] + \sum_{j=1}^{k-1} 2^{-j}(E[B_{j,k-1}] - E[B_{j-1,k-1}]) \\ &= \sum_{j=0}^{k-2} (2^{-j} - 2^{-(j+1)}) E[B_{j,k-1}] \quad + 2^{-(k-1)} E[B_{k-1,k-1}] \\ &= \sum_{j=0}^{k-2} 2^{-(j+1)} E[B_{j,k-1}] \quad + \quad 2^{-(k-1)} E[B_{k-1,k-1}] \, . \end{aligned}$$

□

Note that $E[B_{k,k}] = 2^k - 1/2$. We now give an upper bound for $E[B_{j,k}]$.

Lemma 2.4 (a) *For $0 < j < k$,*

$$B_{j,k} \leq B_{j-1,k-1} + \sum_{t \in L_j} V_t \, ,$$

where V_t is the minimum value of the initial contents $A[s]$ over all 2^{k-j} descendant nodes s in L_k of node t.
(b) *For $0 \leq j < k$,*

$$E[B_{j,k}] \leq \frac{2^{j+1} - 1}{2^{k-j} + 1} \, .$$

Proof (a) If we cut the tree above L_j before bubbling up L_k then $B_{j,k}$ would be no larger, since any movement across the cut would decrease the sum of the values in the top levels. Thus

$$\begin{aligned} B_{j,k} &\leq \text{'}B_{j,k} \text{ with cut'} \\ &= B_{j-1,k-1} + \sum_{t \in L_j} \min(A[t], V_t) \, , \end{aligned}$$

where $A[t]$ is the contents of node t when we are about to bubble up L_k.

(b) By part (a),

$$E[B_{j,k}] \leq E[B_{j-1,k-1}] + 2^j/(2^{k-j} + 1) \, .$$

Thus

$$E[B_{j,k}] \le E[B_{0,k-j}] + \sum_{i=1}^{j} 2^i/(2^{k-j}+1) \le (2^{j+1}-1)/(2^{k-j}+1)\ . \qquad \square$$

Lemma 2.5 *For all k, $y_k < 1+\phi$, where $\phi = \sum_{i\ge 1} 1/(2^i+1) \approx 0.7645$.*

Proof By Lemmas 2.3(b) and 2.4(b),

$$y_k \ \le\ 2^{-(k-1)}\frac{1}{2}(2^k-1) + \sum_{j=0}^{k-2} 2^{-(j+1)}\frac{2^{j+1}-1}{2^{k-1-j}+1} < 1+\phi. \qquad \square$$

The preceding lemma is essentially the upper bound of [BS],[F] in (1.1).

Lemma 2.6 $x_k > x_{k-1} + 2^{-k}\ln 2$ *for $k \ge 1$.*

Proof When bubbling up the nodes in L_k, the expected number of swaps along the two links incident with the root is $\sum_{j=2^k}^{2^{k+1}-1} 1/j > \ln 2$. The expected number of swaps along all other links is at least $2E[X_{k-1}]$ by Lemma 2.2. Thus $E[X_k] > 2E[X_{k-1}] + \ln 2$, and the lemma follows. $\square$

Now $x_0 = 0, x_1, x_2, \ldots$ increase (strictly). But by Lemma 2.5 they are bounded above by $(1+\phi)$, so there exists a constant ω such that $x_k \to \omega$ as $k \to \infty$.

Define $S_k = \sum_{j=0}^{k} X_j$ to be the total number of swaps for the levels up to k, and let $s_k = E[S_k]/(2^{k+1}-1)$. Note that $s_0 = 0$ and for $k \ge 1$,

$$s_k = s_{k-1}\frac{2^k-1}{2^{k+1}-1} + x_k\frac{2^k}{2^{k+1}-1}.$$

It follows that $s_k < x_k$ for $k \ge 1$, that the s_k increase (strictly) with k, and that $s_k \to \omega$ as $k \to \infty$.

So far we have shown that part (a) of Theorem 1.3 holds for perfect heaps. Now let $W_n = \sum_{t=1}^{n} X(t)$ (in particular, $W_{2^{k+1}-1} = S_k$).

Lemma 2.7 $E[W_n]/n \to \omega$ *as* $n \to \infty$.

Proof Let $0 < \varepsilon < \omega$. Choose k_0 such that $\omega - \varepsilon < s_{k_0} (< \omega)$ and thus also $\omega - \varepsilon < x_{k_0} < \omega$. Let $k = \lfloor \log_2(n+1) \rfloor - 1$, so that L_k is the last complete level. Let n be sufficiently large that $k \geq k_0$. We shall show that

$$(\omega - \varepsilon)(n - 2^{k_0}) < E[W_n] < (\omega + \varepsilon)(n + 2^{k_0}),$$

which will establish the lemma.

If $n = 2^{k+1} - 1$ then $E[W_n] = n s_k$ and we are done, since $\omega - \varepsilon < s_k < \omega$. So we may assume that $n \geq 2^{k+1}$. Consider the leaves in level L_{k+1} and their ancestors in level L_{k+1-k_0}. Let $m = \lfloor (n - (2^{k+1} - 1))/2^{k_0} \rfloor$. By Lemma 2.2

$$\begin{aligned} E[\sum_{t=2^{k+1}}^{n} X(t)] &\geq E[\sum_{t=2^{k+1}}^{2^{k+1}-1+m2^{k_0}} X(t)] \geq m 2^{k_0} x_{k_0}, \quad \text{and so} \\ E[W_n] &= E[S_k] + E[\sum_{t=2^{k+1}}^{n} X(t)] \\ &\geq (2^{k+1} - 1) s_k + m 2^{k_0} x_{k_0} \\ &\geq (\omega - \varepsilon)(2^{k+1} - 1 + m 2^{k_0}) \\ &\geq (\omega - \varepsilon)(n - (2^{k_0} - 1)) \end{aligned}$$

as required.

We obtain an upper bound on $E[W_n]$ similarly. Now let $m = \lfloor (2^{k+2} - 1 - n)/2^{k_0} \rfloor$. Then

$$\begin{aligned} E[W_n] &= E[S_{k+1}] - E[\sum_{t=n+1}^{2^{k+2}-1} X(t)\,] \\ &\leq (2^{k+2} - 1) s_{k+1} - m 2^{k_0} x_{k_0} \\ &\leq (\omega + \varepsilon)(n + 2^{k_0} - 1). \end{aligned}$$

□

We have now completed part (a) of the proof of Theorem 1.3. We next present results that will be needed to prove the rest of the theorem.

Lemma 2.8 *For each fixed $j \geq 0$, $a_{k-j,k}$ increases to a limit α_j as $k \to \infty$, where $\sum_{j \geq 0} 2^{-j} \alpha_j = 1$.*

Proof By Lemma 2.2, if $k \leq k'$ then $a_{k-j,k} \leq a_{k'-j,k'}$. Thus $a_{k-j,k}$ increases to a limit α_j (≤ 1) as $k \to \infty$. Further,

$$\sum_{j=0}^{k} 2^{-j} a_{k-j,k} \;=\; 2^{-k} E[\sum_{j=0}^{k} A_{k-j,k}] \;=\; 1 - 2^{-k-1}$$

and it follows easily that $\sum_{j\geq 0} 2^{-j}\alpha_j = 1$. □

From Lemma 2.3(b) and the above lemma the y_k increase (strictly) with k. But we saw in Lemma (2.5) that they are bounded above. Hence y_k tends to the limit $\alpha = \sum_{j\geq 0}\alpha_j$ as $k \to \infty$. Indeed, we may argue as in Lemma 2.6 that

$$E[Y_k] \geq 2E[Y_{k-1}] + |L_k|2^{-k} = 2E[Y_{k-1}] + 1,$$

and so $y_k \geq y_{k-1} + 2^{-k}$.

Finally, in this section we consider 'blocking'. Let s and t be nodes in L_k with $s < t$, and let v be their nearest common ancestor. When node s is bubbled up, the number of nodes i on the insertion path $IP(t)$ with $A[i] > A[t]$ stays the same if $A[s] > A[t]$ or $A[t] > A[v]$, and decreases by 1 if $A[s] < A[t] < A[v]$. Let us then say that s *blocks* t if, when s is about to be bubbled up we have $A[s] < A[t] < A[v]$. Set $Z(s,t) = 1$ if s blocks t and $Z(s,t) = 0$ otherwise. Observe that for each $t \in L_k$,

$$X(t) = Y(t) - \sum\{Z(s,t) : s \in L_k, s < t\}.$$

Hence $X_k = Y_k - Z_k$, where $Z_k = \sum\{Z(s,t) : s,t \in L_k, s < t\}$.

Let $0 \leq j \leq k$. We wish to compare Z_k with the corresponding quantity, which we call Z'_k, obtained by cutting the tree above level L_{k-j}. By Lemma 2.2(b), if s blocks t in the cut tree then this also happens in the original tree. Hence always $Z_k \geq Z'_k$. But Z'_k is the sum of 2^{k-j} independent random variables each distributed like Z_j. Thus in particular, $E[Z_k] \geq 2E[Z_{k-1}]$ and so the quantities $z_k = 2^{-k}E[Z_k]$ are non-decreasing. But $z_k \leq y_k \leq \alpha$ and so z_k tends to some limit β as $k \to \infty$.

We can in fact prove that the z_k increase strictly, in the spirit of Lemma 2.6 (though this yields only a small improvement in our estimates).

Lemma 2.9 $z_k \geq z_{k-1} + 2^{-k}(1/6)$.

Proof Denote the left half of the nodes in L_k by L and the right half by R. Thus $L = \{2^k, \ldots, 2^k + 2^{k-1} - 1\}$ and $R = \{2^k + 2^{k-1}, \ldots, 2^{k+1} - 1\}$. Let $T = \bigcup_{j=0}^{k-1} L_j$. Let

$$\begin{aligned} Z_L &= \sum\{Z(s,t) : s,t \in L, s < t\} \\ Z_R &= \sum\{Z(s,t) : s,t \in R, s < t\} \\ Z_{L,R} &= \sum\{Z(s,t) : s \in L, t \in R\} \end{aligned}$$

Thus $Z_k = Z_L + Z_R + Z_{L,R}$.

Then arguing as above, both $E[Z_L]$ and $E[Z_R]$ are at least $E[Z_{k-1}]$. Further

$$E[Z_{L,R}] = \sum_{t \in R} \text{Prob}\{Z(s,t) = 1 \text{ for some } s \in L\}.$$

Fix a node t in R. Then $\text{Prob}\{Z(s,t) = 1 \text{ for some } s \in L\}$

$$\begin{aligned}
&= \text{Prob}\ \{\min_{s \in L} A[s] < A[t] < \min_{x \in T} A[x]\} \\
&= \text{Prob}\ \{\min_{s \in L} A[s] < \min_{x \in T \cup \{t\}} A[x] \text{ and } A[t] < \min_{x \in T} A[x]\} \\
&= \text{Prob}\ \{\min_{s \in L} A[s] < \min_{x \in T \cup \{t\}} A[x]\}\ \text{Prob}\ \{A[t] < \min_{x \in T} A[x]\} \\
&= \frac{|L|}{|L| + |T| + 1} \times \frac{1}{|T| + 1} = \frac{1}{3 \cdot 2^k}.
\end{aligned}$$

Hence $E[Z_{L,R}] = 1/6$ and we are done. □

3 An algorithm for computing $E[W_n]$

Recall that $E[W_n]$ is the expected number of swaps performed by Williams' algorithm in the construction of a heap with n keys. It is trivial to compute $E[W_n]$ by considering all $n!$ possible linear orderings of the input keys. In this section we present an algorithm to compute $E[W_n]$ in $O(n^3)$ time and $O(n^2)$ space.

We wish to compute $E[W_n]$ as a function of $E[W_{n-1}]$. To that end, we introduce some notation. The *rank* of a number x in a set of numbers is its placement in the ordered set; thus the smallest number has rank 1, the next smallest rank 2, etc. Let $\overline{A}_n[i]$ be the rank of $A[i]$ among $A[1, \ldots, n]$ after exactly n keys have been inserted. For $1 \leq i, j \leq n$, define $P(n, i, j)$ as $\text{Prob}\{\overline{A}_n[i] = j\}$, i.e. the probability that after exactly n keys have been inserted, the key currently in node i is the j^{th} smallest of the n keys. For example, $P(n, 1, 1) = 1$ for $n \geq 1$, since $A[1, \ldots, n]$ forms a heap. Finally, define $I(n)$ as the expected number of swaps to insert the n^{th} key, that is $I(n) = E[X(n)]$. Clearly,

$$E[W_n] = E[W_{n-1}] + I(n)\ .$$

The main result of this section is that given $P(n-1 \cdot \cdot)$ it is possible to compute both $P(n \cdot \cdot)$ and $I(n)$.

We first show how to compute $I(n)$ from $P(n-1 \cdot \cdot)$. Recall that the insertion path $IP(t)$ from node t is the set of nodes $\{\lfloor t/2^j \rfloor : j = 1, \ldots, \lfloor \lg t \rfloor\}$.

Define the *extended insertion path* $IP^+(t)$ as $IP(t) \cup \{t\}$. Let $R(n)$ denote the (initial) rank of the n^{th} key among the first n keys. Observe that during the insertion of some key with $R(n) = k$, a swap takes place at node i of the insertion path if and only if $k \leq \overline{A}_{n-1}[i]$. Assuming that $\overline{A}_{n-1}[i] = j$, a swap will occur at node i if and only if $R(n) \leq j$. Thus

$$I(n) = \frac{1}{n} \sum_{i \in IP(n)} \sum_{j=1}^{n-1} j \, P(n-1, i, j) \; .$$

We now show how to compute $P(n \cdot\cdot)$ from $P(n-1 \cdot\cdot)$. We organize the terms that contribute to $P(n \cdot\cdot)$ according to the n^{th} key inserted. Define $P(n, i, j, k)$ as $\text{Prob}\{\overline{A}_n[i] = j \,|R(n) = k\}$. Observe that

$$P(n, i, j) = \frac{1}{n} \sum_{k=1}^{n} P(n, i, j, k) \; .$$

Note that if $R(n) = k$, then the ranks of all keys that had rank k or greater before insertion are incremented by one. Thus, for a node i not on $IP^+(n)$,

$$P(n, i, j, k) = \begin{cases} P(n-1, i, j-1) & \text{if } k < j \\ 0 & \text{if } k = j \\ P(n-1, i, j) & \text{if } k > j \end{cases}$$

and so

$$n \cdot P(n, i, j) = (j-1)P(n-1, i, j-1) \; + \; (n-j)P(n-1, i, j) \; .$$

Here and for the rest of this section, array indices are understood to be integers; any fraction x/y used an array index is actually $\lfloor x/y \rfloor$.

Define $E(n, i, k)$ as the probability a key with $R(n) = k$ ends at node i when inserted. Obviously, $E(n, i, k) = 0$ if node i is not on $IP^+(n)$. If i is on $IP^+(n)$, then the key stops at node i if and only if $\overline{A}_{n-1}[i/2] < k$ and $\overline{A}_{n-1}[i] \geq k$. Thus

$$\begin{aligned} E(n, i, k) &= \text{Prob}\{\overline{A}_{n-1}[i/2] < k \text{ and } \overline{A}_{n-1}[i] \geq k\} \\ &= \text{Prob}\{\overline{A}_{n-1}[i/2] < k\} - \text{Prob}\{\overline{A}_{n-1}[i/2] < k \text{ and } \overline{A}_{n-1}[i] < k\} \; . \end{aligned}$$

Since the contents of $A[1, \ldots, n-1]$ form a heap before insertion of the n^{th} key, $\overline{A}_{n-1}[i/2] < \overline{A}_{n-1}[i]$. Thus both $\overline{A}_{n-1}[i/2] < k$ and $\overline{A}_{n-1}[i] < k$ if

and only if $\overline{A}_{n-1}[i] < k$. It follows that

$$\begin{aligned} E(n,i,k) &= \text{Prob}\{\overline{A}_{n-1}[i/2] < k\} - \text{Prob}\{\overline{A}_{n-1}[i] < k\} \\ &= \sum_{t<k} P(n-1,i/2,t) - \sum_{t<k} P(n-1,i,t) \\ &= \sum_{t=1}^{k-1} (P(n-1,i/2,t) - P(n-1,i,t)) . \end{aligned}$$

Thus for a node i on $IP^+(n)$,

$$P(n,i,j,k) = \begin{cases} P(n-1,i/2,j-1) & \text{if } k < j \\ E(n,i,k) & \text{if } k = j \\ P(n-1,i,j) & \text{if } k > j \end{cases}$$

and so

$$n \cdot P(n,i,j) = (j-1)P(n-1,i/2,j-1) \ + \ (n-j)P(n-1,i,j) \ + \ E(n,i,j) .$$

We use the above recurrences to compute $E[W_n]$. For each node i not on $IP^+(n)$, the n values $P(n,i,\ \cdot)$ can be computed in $O(n)$ time (arithmetic operations); for i on $IP^+(n)$, these values can also be computed in $O(n)$ time, by saving the value of $E(n,i,k-1)$ for the computation of $E(n,i,k)$. Thus $E[W_n]$ can be computed in $O(n^3)$ time, using $O(n^2)$ space. Finally, by storing the probability arrays $P(n,i,\ \cdot)$ only for nodes i on $IP^+(n)$, it is possible to compute $E[W_n]$ in only $O(n \lg n)$ space but $O(n^3 \lg n)$ time.

To conclude the section, we point out that just as there is a recurrence relation involving the values $P(n\ \cdot\ \cdot)$, so there are recurrence relations involving weighted sums of $P(n\ \cdot\ \cdot)$. Since only the value $\sum jP(n,i,j)$ is needed in computing $I(n)$, it is natural to ask if $\sum jP(n,i,j)$ can be computed from $\sum jP(n-1,i,j)$ and $\sum jP(n-1,i/2,j)$ directly. Define $S(n,i,t)$ as $\sum_{j=1}^{n} \binom{j}{t} P(n,i,j)$. Straightforward algebraic manipulation, and a little perseverence, yield the following results.

For a node i not on $IP^+(n)$,

$$n \cdot S(n,i,t) = (\frac{t-1}{n})S(n-1,i,t-1) + (1+\frac{t}{n})S(n-1,i,t) ,$$

and for a node i on $IP^+(n)$,

$$\begin{aligned} n \cdot S(n,i,t) &= (t-1)S(n-1,i/2,t-1) \\ &+ (2t-1)S(n-1,i/2,t) \ + \ t \cdot S(n-1,i/2,t+1) \\ &+ \begin{cases} \binom{n+1}{t+1} & \text{if } i = n \\ (n+1-t)S(n-1,i/2,t) - t \cdot S(n-1,i,t+1) & \text{if } i < n . \end{cases} \end{aligned}$$

4 Bounds for Williams' heaps construction constant ω

In this section we establish the bounds on ω given in Theorem 1.3(b). A lower bound for ω comes directly from Lemma 2.6. For any k,

$$\omega > x_k + \ln 2 \sum_{j \geq k+1} 2^{-j} = x_k + 2^{-k} \ln 2$$

The value $k = 11$ gives the stated bound.

We obtain the upper bound for $\omega = \alpha - \beta$ by giving an upper bound for α and a lower bound for β. The lower bound for β is easy. From Lemma 2.3(b) and the values $a_{j,10}$ in Appendix 2,

$$y_{11} = \sum_{j=0}^{10} a_{j,10} = 1.60\ 10\ 73\ 11\ 68\ .$$

But by Lemma 2.9

$$\begin{aligned} \beta &\geq z_{11} + 2^{-11}(1/6) \\ &= y_{11} - x_{11} + 2^{-11}(1/6) = 0.32\ 36\ 51\ 64\ 70\ . \end{aligned}$$

Establishing an upper bound for α requires several steps. Begin with observing that by Lemma 2.3(b)

$$y_k = \sum_{j=0}^{k-1} a_{j,k-1} = 1 - 2^{-k} + 2^{-k} \sum_{j=1}^{k-1} 2^j E[B_{k-1-j,k-1}]\ .$$

We shall use two different upper bounds for the terms $E[B_{k-1-j,k-1}]$ for $j \geq 1$. By Lemma 2.4 we have

$$2^{-k} E[B_{k-1-j,k-1}] < \frac{2^{-j} - 2^{-k}}{2^j + 1} < \frac{1}{2^j(2^j+1)} \quad \text{for } 1 \leq j \leq k-1\ .$$

Also, however, we have

$$\begin{aligned} 2^{-k} E[B_{k-1-j,k-1}] &= 2^{-k} E[B_{k-1,k-1}] - 2^{-k} \sum_{i=0}^{j-1} E[A_{k-1-i,k-1}] \\ &= \frac{1}{2} - 2^{-k-1} - \sum_{i=0}^{j-1} 2^{-i-1} a_{k-1-i,k-1}\ . \end{aligned}$$

But by Lemma 2.8, $a_{k-1-i,k-1} \geq a_{11-i,11}$ once $k \geq 12$. Hence, for $1 \leq j \leq 12$

$$2^{-k}E[B_{k-1-j,k-1}] < \frac{1}{2} - \sum_{i=0}^{j-1} 2^{-i-1} a_{11-i,11}\ , = b_j \ \text{ say.}$$

These bounds are displayed in Appendix 2. We use the latter bound for the last few levels from the bottom and the former bound for the remaining levels. Thus we have

$$\begin{aligned} y_k &< 1 + \sum_{j=1}^{5} 2^j b_j + \sum_{j=6}^{k-1} \frac{1}{2^j+1} \\ &< 1 + \phi - \sum_{j=1}^{5} (\frac{1}{2^j+1} - 2^j b_j)\ . \end{aligned}$$

Here

$$1 + \phi = 1 + \sum_{j \geq 1} \frac{1}{2^j+1} = 1.76\ 44\ 99\ 78\ ,$$

as in Lemma 2.5. Also

$$\sum_{j=1}^{5} (\frac{1}{2^j+1} - 2^j b_j) = 0.14\ 13\ 75\ 69\ 36\ .$$

Hence

$$\alpha \leq 1.62\ 31\ 24\ 08\ 67\ .$$

We now have an upper bound on α and a lower bound on β and thus obtain

$$\omega = \alpha - \beta \leq 1.29\ 94\ 72\ 43\ 96\ .$$

Finally, some comments on the accuracy of our bounds. We do not need a lower bound on α, but it is interesting to see how close we are. Using Lemma 2.3(b) we may calculate y_{12} from the table in Appendix 2. Then by the comments following Lemma 2.8

$$\alpha \geq y_{12} + 2^{-12} = 1.60\ 39\ 01\ 46\ 05.$$

Also,

$$\begin{aligned} \beta &= \alpha - \omega \leq 1.62\ 31\ 25\ 08\ 67\ -\ 1.29\ 94\ 72\ 43\ 96 \\ &= 0.34\ 52\ 82\ 78\ 60\ . \end{aligned}$$

Thus the ranges for ω, α, β are each about 0.02.

5 Probability bounds

In this section we shall prove part (c) of Theorem 1.3. We follow roughly the route taken in section 2 for investigating $E[W_n]$. We show that W_n/n is unlikely to be much below ω; that $2^{-k}Z_k$ is unlikely to be much below β; that $2^{-k}Y_k$ is unlikely to be much above α; and thus finally that W_n/n is unlikely to be much above ω.

We shall make repeated use of the following inequalities from Hoeffding [H], see also McDiarmid [M]. Let $T_1, T_2, \ldots, T_n$ be independent random variables with $0 \le T_i \le 1$ for each i. Let $\overline{T} = (1/n)\sum_{i=1}^{n} T_i$, and $E[\overline{T}] = \mu$. Then for any $t > 0$

$$\text{Prob}\{\ |\overline{T} - \mu|\ \ge t\} \le 2\exp(-2nt^2)\ . \tag{5.1}$$

Also, if $0 < \varepsilon \le 1$ then

$$\text{Prob}\{\overline{T} \ge (1+\varepsilon)\mu\} \le \exp(-\varepsilon^2 n\mu/3)\ . \tag{5.2}$$

Lemma 5.3 *For any $\varepsilon > 0$ there exists $\delta > 0$ such that* $\text{Prob}\{W_n/n < \omega - \varepsilon\} < \exp(-\delta n)$.

Proof Recall that $s_k = E[\sum_{j=0}^{k} X_j]/(2^{k+1}-1)$ and that s_k tends to w as $k \to \infty$. Choose k_0 such that $s_{k_0} > \omega - \varepsilon/2$. We may cover all but $O(\lg n)$ nodes in a heap on n nodes with disjoint k_0-heaps. Thus by Lemma 2.2 W_n is at least the sum Σ_n of $n/(2^{k_0+1}-1) - O(\lg n)$ independent random variables, each distributed like S_{k_0}. Now $0 \le S_{k_0} \le k_0 2^{k_0+1}$ and

$$E[S_{k_0}] = (2^{k_0+1}-1)s_{k_0} > (2^{k_0+1}-1)(\omega - \varepsilon/2)\ .$$

So by inequality (5.1)

$$\begin{aligned} \text{Prob}\{W_n/n < \omega - \varepsilon\} & \\ & \le \text{Prob}\{\Sigma_n/n < \omega - \varepsilon\} \\ & \le \exp(-(1+o(1))(\varepsilon^2/2)n)\ . \end{aligned}$$

□

Lemma 5.4 *For any $\varepsilon > 0$ there exists $\delta > 0$ such that* $\text{Prob}\{2^{-k}Z_k \le \beta - \varepsilon\} \le \exp(-\delta 2^k)$.

Proof Recall that $Z_k = Y_k - X_k$ (so that $Z_k \le Y_k \le k2^k$) and that $z_k = 2^{-k}E[Z_k] \to \beta$ as $k \to \infty$. Choose k_0 such that $z_{k_0} \ge \beta - \varepsilon/2$. Then

the random variable Z_{k_0} satisfies $0 \leq Z_{k_0} \leq k_0 2^{k_0}$ and $E[Z_0] = 2^{k_0} z_{k_0}$. Also, as we saw before Lemma 2.9, Z_k is distributed at least as the sum of 2^{k-k_0} independent copies of Z_{k_0}. Hence by inequality (5.1)

$$\begin{aligned} \mathrm{Prob}\{2^{-k} Z_k < \beta - \varepsilon\} & \\ & \leq \mathrm{Prob}\{2^{-(k-k_0)} Z_k < 2^{k_0} z_{k_0} - 2^{k_0-1}\varepsilon\} \\ & \leq 2 \cdot \exp(2^{k-k_0+1}(\varepsilon/2k_0)^2) \,. \end{aligned}$$

□

Lemma 5.5 *For any $\varepsilon > 0$, for each integer $j \geq 0$, there exists $\delta > 0$ such that*
$\mathrm{Prob}\{\mid 2^{-(k-j)} A_{k-j,k} - \alpha_j \mid \; > \varepsilon\} < \exp(-\delta 2^k)$.

Proof Let $\varepsilon > 0$ and $j \geq 0$. Let $k_0 > j$ be such that $a_{k_0-j,k_0} > \alpha_j - \varepsilon/2$, and consider $k \geq k_0$. By Lemma 2.2, $A_{k-j,k}$ is in distribution at least the sum of 2^{k-k_0} independent copies of A_{k_0-j,k_0}. Thus $2^{-(k-j)} A_{k-j,k}$ is in distribution at least the average of 2^{k-k_0} independent copies of $2^{-(k_0-j)} A_{k_0-j,k_0}$; these random variables are bounded between 0 and 1, and have mean $a_{k_0-j,k_0} > \alpha_j - \varepsilon/2$. Hence by inequality (5.1),

$$\mathrm{Prob}\{2^{-(k-j)} A_{k-j,k} < \alpha_j - \varepsilon\} < 2 \exp(-2^{k-k_0-1}\varepsilon^2) \,.$$

This gives us half of the lemma.

Again let $\varepsilon > 0$ and $j \geq 0$. Now let $k_0 > j$ be such that $\sum_{i=0}^{k_0} 2^{-i}\alpha_i \geq 1 - \varepsilon 2^{-j-2}$, and consider $k \geq k_0$. Suppose that $2^{-(k-j)} A_{k-j,k} \geq \alpha_j + \varepsilon$. If for each $i = 0, \ldots, k_0$ $i \neq j$ we have

$$2^{-(k-i)} A_{k-i,k} \geq \alpha_i - \varepsilon 2^{-j-2} \,,$$

then

$$\begin{aligned} 2^{-k} B_{k,k} & \geq \sum_{i=0}^{k_0} 2^{-i}\, 2^{-(k-i)} A_{k-i,k} \\ & \geq \sum_{i=0, i\neq j}^{k_0} 2^{-i}(\alpha_i - \varepsilon 2^{-j-2}) + 2^{-j}(\alpha_j + \varepsilon) \\ & \geq \sum_{i=0}^{k_0} 2^{-i}\alpha_i + \varepsilon 2^{-j-1} \\ & \geq 1 + \varepsilon 2^{-j-2} \,. \end{aligned}$$

Hence

$$\begin{aligned}\mathrm{Prob}\{2^{-(k-j)}A_{k-j,k} \geq \alpha_j + \varepsilon\} \quad &\leq \quad \sum_{i=0, i\neq j}^{k_0} \mathrm{Prob}\{2^{-(k-i)}A_{k-i,k} < \alpha_i - \varepsilon 2^{-j-2}\} \\ &\quad + \mathrm{Prob}\{2^{-k}B_{k,k} \geq 1 + \varepsilon 2^{-j-2}\}\,.\end{aligned}$$

Finally, we may use the half of the lemma already proved to handle the sum above, and one further application of inequality (5.1) to handle the last term. □

Lemma 5.6 *For any $\varepsilon > 0$,* $\mathrm{Prob}\{|2^{-k}Y_k - \alpha| > \varepsilon\} = o(\exp(-2^{k-1}/k^2(\lg k)^8))$.

Proof Recall Lemma 2.3, and note that conditional on the values $(A[i] : i = 1, \ldots, 2^k - 1)$ at the time when we are about to bubble-up level L_k, the random variables $(Y(t) : t \in L_k)$ are independent. Let E_k and F_k be the events

$$\begin{aligned}E_k &= \{\,|\,2^{-k}Y_k - \alpha\,|\, > 2\varepsilon\}\,, \quad \text{and} \\ F_k &= \{\,|\,\sum_{j=0}^{k-1} 2^{-j}A_{j,k-1} - \alpha\,|\, > \varepsilon\}\,.\end{aligned}$$

We shall use the inequality

$$\mathrm{Prob}(E_k) \leq \mathrm{Prob}(E_k \mid \overline{F}_k) + \mathrm{Prob}(F_k)\,.$$

Since each $0 \leq Y(t) \leq k$, inequality (5.1) applied to the $Y(t)/k$ yields

$$\mathrm{Prob}(E_k \mid \overline{F}_k) \leq 2\exp(-2^{k+1}(\varepsilon/k)^2)\,.$$

Hence to prove the lemma it will suffice to show that

$$\mathrm{Prob}(F_{k+1}) = o(\exp(-2^{k-1}/k^2(\lg k)^8)\,)\,. \tag{5.7}$$

(It is tidier to replace k by $k+1$.)

To do this we split the sum $\sum_{j=0}^{k} 2^{-j}A_{j,k}$ into parts. We shall choose a suitably large integer constant k_1 and set $k_2 = \lfloor \lg k + 3\lg\lg k \rfloor$; and write

$$\begin{aligned}\sum_{j=0}^{k} 2^{-j}A_{j,k} &= \sum_{j=0}^{k-k_2} + \sum_{j=k-k_2+1}^{k-k_1} + \sum_{j=k-k_1+1}^{k} \\ &= \sum\nolimits_1 + \sum\nolimits_2 + \sum\nolimits_3\,.\end{aligned}$$

(We are assuming that k is sufficiently large that $k_2 > k_1$.) We shall see that the contributions from Σ_1 and Σ_2 are negligible. By Lemma 5.5, if the constant k_1 is sufficiently large that $\sum_{j=0}^{k_1-1} \alpha_j \geq \alpha - \varepsilon/4$, then there exists $\delta > 0$ such that

$$\text{Prob}\{ \mid \Sigma_3 - \alpha \mid \ > \varepsilon/3\} < \exp(-\delta 2^k)\,. \tag{5.8}$$

First let us dispose of the sum Σ_1. In a heap with $2^{k+1}-1$ nodes the contents $A[t]$ of a node t in level L_j is at most the $(j+1)^{\text{st}}$ smallest element originally in the path from the root to t together with the corresponding subtree rooted at t, and this set contains $2^{k-j+1}+j-1 \geq 2^{k-j}+j$ nodes. Thus

$$\begin{aligned} \text{Prob}\{A[t] \geq \delta\} &\leq \binom{2^{k-j}+j}{j}(1-\delta)^{2^{k-j}} \\ &\leq (2^{k-j}+1)^j \exp(-\delta 2^{k-j})\,. \end{aligned}$$

Now if $A_{j,k} \geq \delta 2^{j+1}$ then $A[t] \geq \delta$ for at least $\delta 2^j$ nodes t in L_j. Hence

$$\begin{aligned} \text{Prob}\{A_{j,k} \geq \delta 2^{j(k-j+1)}\} &\leq 2^{2^j}(2^{j(k-j+1)}\exp(-\delta 2^{k-j})\,)^{\delta 2^j} \\ &\leq \exp(2^j \ln 2 - \delta^2 2^{k-1}) \end{aligned}$$

as long as

$$\delta 2^{k-j} \geq 2j(k-j+1)\ln 2\,. \tag{5.9}$$

For $0 \leq j \leq k - 4\lg k$ take $\delta = \varepsilon/(12k)$. Then (5.9) holds (for k sufficiently large). Hence

$$\text{Prob}\{\sum_{j=0}^{\lfloor k-4\lg k\rfloor} 2^{-j}A_{j,k} \geq \varepsilon/6\} < k\exp(2^{k-4\lg k}\ln 2 - \delta^2 2^{k-1}) = o(\exp(-\frac{\varepsilon^2}{289}\frac{2^k}{k^2})).$$

For $k - 4\lg k < j \leq k - k_2$ take $\delta = \varepsilon/(48\lg k)$. Again (5.9) holds, and so

$$\begin{aligned} \text{Prob}\{\sum_{j=\lceil k-4\lg k\rceil}^{k-k_2} 2^{-j}A_{j,k} \geq \varepsilon/6\} &\leq (4\lg k)\exp(2^{k-\lg k}\ln 2 - \delta^2 2^{k-1}) \\ &= o(\exp(-\frac{\varepsilon^2}{4609}\frac{2^k}{(\lg k)^2})\,)\,. \end{aligned}$$

Hence

$$\text{Prob}\{\Sigma_1 \geq \varepsilon/3\} = o(\exp(-\frac{\varepsilon^2}{290}\frac{2^k}{k^2})\,)\,. \tag{5.10}$$

It remains to dispose of the sum Σ_2. In Lemma 2.4(a) the 2^j random variables V_t are independent, $0 \le V_t \le 1$, and $E[V_t] = 1/(2^{k-j}+1)$. Hence by inequality (5.2)

$$\text{Prob}\{2^{-j}\sum_{t\in L_j} V_t > 2\cdot 2^{-(k-j)}\} < \exp(-2^j 2^{-(k-j)}/3)$$

and so

$$\text{Prob}\{B_{j,k} - B_{j-1,k-1} > 2^{2j-k+1}\} < \exp(-2^{2j-k-2})\,.$$

Thus also

$$\text{Prob}\{B_{j-i,k-i} - B_{j-i-1,k-i-1} > 2^{2j-k-i+1}\} < \exp(-2^{2j-k-i-2})\,.$$

Hence

$$\text{Prob}\{B_{j,k} - B_{j-s,k-s} > 2^{2j-k+2}\} < s\exp(-2^{2j-k-s-2})\,.$$

Now let $s = \lfloor 2\lg\lg k\rfloor$. Then for $j \ge k - k_2 + 1$,

$$\begin{aligned} 2j-k-s-2 &\ge k - 2k_2 - s \\ &\ge k - 2\lg k - 8\lg\lg k\,. \end{aligned}$$

Also of course $B_{j-s,k-s} \le 2^{j-s+1}-1$. Hence for $j \ge k-k_2+1$,

$$\text{Prob}\{2^{-j}B_{j,k} > 2^{j-k+2} + 2^{-s+1}\} \le (2\lg\lg k)\exp(-\frac{2^k}{k^2(\lg k)^8})\,.$$

Now $\Sigma_2 \le \sum_{j=k-k_2+1}^{k-k_1} 2^{-j}B_{j,k}$, so

$$\text{Prob}\{\Sigma_2 > 2^{-k_1+3} + (k_2-k_1)2^{-s+1}\} \le (k_2-k_1)(2\lg\lg k)\exp(-\frac{2^k}{k^2(\lg k)^8})\,.$$

But $2^{-k_1+3} \le \varepsilon/4$ (if we choose k_1 large enough) and $(k_2-k_1)2^{-s+1} = o(1)$. So

$$\text{Prob}\{\Sigma_2 \ge \varepsilon/3\} = o(\exp(-\frac{2^{k-1}}{k^2(\lg k)^8})\,)\,.$$

This last inequality together with (5.8) and (5.10) yields (5.7). This completes the proof of Lemma 5.6. □

We can now handle perfect heaps. We are almost home.

Lemma 5.11 *For any* $\varepsilon > 0$, $\mathrm{Prob}\{S_k \geq (\omega+\varepsilon)(2^{k+1}-1)\} = o(\exp(-2^k/k^3(\lg k)^9)$.

Proof Lemmas 5.4 and 5.6 show that

$$\mathrm{Prob}\{X_k \geq (\omega+\varepsilon)2^k\} = o(\exp(-2^{k-1}/k^2(\lg k)^8)\ .$$

Let $n = 2^{k+1}-1$. The first $\varepsilon n/2k$ nodes t can contribute at most $\varepsilon n/2$ to S_k. Let $k_0 = \lfloor \lg(\varepsilon n/2k) \rfloor = k - \lg k + O(1)$. Then

$$\begin{aligned} & \mathrm{Prob}\{S_k \geq (\omega+\varepsilon)n\} \\ \leq\ & \mathrm{Prob}\{\sum_{j=k_0}^{k} X_j \geq (\omega+\varepsilon/2)\sum_{j=k_0}^{k} 2^j\} \\ =\ & o(\ (k-k_0+1)\exp(-\frac{2^{k_0-1}}{k^2(\lg k)^8})\ ) \\ =\ & o(\exp(-\frac{2^k}{k^3(\lg k)^9})\ )\ . \end{aligned}$$

□

Lemma 5.12 *For any* $\varepsilon > 0$, $\mathrm{Prob}\{W_n \geq (\omega+\varepsilon)n\} = o(\exp(-n/(\lg n)^4)\ )$.

Proof Let $2^{k+1} \leq n \leq 2^{k+2}-1$. Now if $W_n \geq (\omega+2\varepsilon)n$ and $S_{k+1} - W_n \geq (2^{k+2}-1-n)\omega - n\varepsilon$ then $S_{k+1} \geq (2^{k+2}-1)(\omega+\varepsilon/2)$. Hence

$$\begin{aligned} \mathrm{Prob}\{W_n \geq (\omega+\varepsilon)n\} \ \leq\ & \mathrm{Prob}\{S_{k+1} \geq (2^{k+2}-1)(\omega+\varepsilon/2)\ \} \\ +\ & \mathrm{Prob}\{S_{k+1} - W_n < (2^{k+2}-1-n)\omega - n\varepsilon\}\ . \end{aligned}$$

By Lemma 5.11 the former term on the right hand side is $o(\exp(-n/(\lg n)^4))$. We may handle the latter term much as in the proof of Lemma 5.3, and we find that for some $\delta > 0$ this term is $o(\exp(-\delta n)\ )$. □

Lemmas 5.3 and 5.12 combine to complete the proof of Theorem 1.3.

6 Repeated insertion into equi-probable heaps

In this section, we consider the simplified approximation of the average case behaviour of Williams' heap construction in which we repeatedly insert a uniform random key into a uniform random heap. Thus we are assuming as in Porter and Simon [PS] that

> each possible heap of the first $n-1$ keys is equally likely, and
> each possible rank of the n^{th} key is equally likely.

We refer to this as the 'equi-probable' model.

Our interest is to see whether results obtained here are similar to those obtained earlier, since one may be tempted to conduct a similar approximate analysis in more complicated situations, for example when considering heapsort (see [C]).

Following our notation of the previous sections, we shall define $\widetilde{X}(t)$ as the number of swaps or promotions to insert the t^{th} key, $\widetilde{W}_n$ as $\sum_{t=1}^{n} \widetilde{X}(t)$, and $\widetilde{I}(n)$ as $E[\widetilde{X}(n)]$. The main result of this section is the following.

Theorem 6.1

(a) There exists a constant $\widetilde{\omega}$ such that $E[\widetilde{W}_n] \to \widetilde{\omega}$ as $n \to \infty$.
(b) $1.27057815\ldots < \widetilde{\omega} < 1.27057824\ldots$
(c) For any $\varepsilon > 0$, $\text{Prob}(|\widetilde{W}_n/n - \widetilde{\omega}| > \varepsilon) = O(\exp(-\varepsilon^2 n/\lg^2 n))$.

Thus $\widetilde{\omega} = 1.270578\ldots$ and is strictly less than ω.

For a heap of size n, let $g(n)$ be the size of the subheap that is obtained by removing the root, and that contains node n. Note that for $l = \lfloor \lg n \rfloor$,

$$g(n) = \begin{cases} n - 2^{l-1} & \text{if } n < 2^l + 2^{l-1} \\ n - 2^l & \text{if } n \geq 2^l + 2^{l-1} \ . \end{cases}$$

Porter and Simon [PS] observed that

$$\widetilde{I}(n) = \frac{\lfloor \lg n \rfloor}{n} + \frac{n-1}{n}\widetilde{I}(g(n)) \ . \tag{6.2}$$

The proof of parts (a) and (b) of the theorem will follow routinely from (6.2). The proof of part (c) will then follow from Hoeffding's inequality (5.1).

Proof Continuing to mimic previous notation, we define $\widetilde{x}_k$ as the average of the expected insertion costs along the k^{th} level L_k, namely $\sum_{t=2^k}^{2^{k+1}-1} \widetilde{I}(t)/2^k$. For a node t on level L_k, note that $k = \lfloor \lg t \rfloor$ and that $g(t + 2^k) = t$ and $g(t + 2^{k+1}) = t$. Thus for $t > 1$,

$$\begin{aligned} \widetilde{I}(t+2^k) + \widetilde{I}(t+2^{k+1}) &= \frac{k+1}{t+2^k} + (1 - \frac{1}{t+2^k})\widetilde{I}(t) \\ &\quad + \frac{k+1}{t+2^{k+1}} + (1 - \frac{1}{t+2^{k+1}})\widetilde{I}(t) \\ &= 2\widetilde{I}(t) + (\frac{1}{t+2^k} + \frac{1}{t+2^{k+1}})(k+1-\widetilde{I}(t)\ ) \ . \end{aligned}$$

Thus the above yields, for all $k \geq 1$

$$\widetilde{I}(t+2^k) + \widetilde{I}(t+2^{k+1}) > 2\widetilde{I}(t) \ ,$$

and so

$$\widetilde{x}_{k+1} > \widetilde{x}_k \ .$$

(See also Lemma 2.6.)

We now establish an upper bound on $\widetilde{x}_{k+1}$.

$$\begin{aligned}
\widetilde{x}_{k+1} &= 2^{-(k+1)} \sum_{t \in L_{k+1}} \widetilde{I}(t) \\
&= 2^{-(k+1)} \sum_{t \in L_k} (\ 2\widetilde{I}(t) + (\frac{1}{t+2^k} + \frac{1}{t+2^{k+1}})(k+1-\widetilde{I}(t)) \ ) \\
&< 2^{-(k+1)} \sum_{t \in L_k} (\ 2\widetilde{I}(t) + (\frac{1}{2^k+2^k} + \frac{1}{2^k+2^{k+1}})(k+1-\widetilde{I}(t)) \ ) \\
&= 2^{-(k+1)} \sum_{t \in L_k} (\ 2\widetilde{I}(t) + \frac{5}{3} \ \frac{1}{2^{k+1}}(k+1-\widetilde{I}(t)) \ ) \\
&= 2^{-k} \sum_{t \in L_k} (\ \widetilde{I}(t) + \frac{5}{6} \ \frac{1}{2^{k+1}}(k+1-\widetilde{I}(t)) \ ) \\
&= \widetilde{x}_k + \frac{5}{6} \ \frac{1}{2^{k+1}}(k+1-\widetilde{x}_k) \\
&< \widetilde{x}_k + \frac{5}{6} \ \frac{k+1}{2^{k+1}} \ .
\end{aligned}$$

Hence, for all $j \geq 1$

$$\widetilde{x}_{k+j} < \widetilde{x}_k + \frac{5}{6} \sum_{i \geq k+1} \frac{i}{2^i} = \widetilde{x}_k + \frac{5}{6} \ \frac{k+2}{2^k} \ .$$

Thus $\widetilde{x}_k \to$ a limit $\widetilde{\omega}$ as $k \to \infty$, and for any k, $\widetilde{x}_k < \widetilde{\omega} < \widetilde{x}_k + \frac{5}{6}\frac{k+2}{2^k}$. We may now show much as in Section 2 that also $E[\widetilde{W}_n]/n \to \widetilde{\omega}$ as $n \to \infty$. This completes our proof of part (a) of the theorem. We take $k = 28$ (see Appendix 2) in the last inequality to obtain an error bound of $9.313\ldots \times 10^{-8}$, which gives part (b). Finally, part (c) follows from part (a) and Hoeffding's inequality (5.1) on noting that $\widetilde{W}_n$ is the sum of n independent random variables bounded between 0 and $\lfloor \lg n \rfloor$. □

7 Concluding remarks

We have described fairly precisely the average case behaviour of Williams' method of constructing a heap by repeated insertion. This complements recent work on the average case analysis of variants of Floyd's method, as mentioned in the introduction. We have also shown that the 'equiprobable' approximation slightly underestimates the number of comparisons required.

The most interesting related open problem is the average case analysis of heapsort (with any of its trickledown variants). While some empirical data (e.g. see [K]) and partial theoretical results (e.g. [C]) are known, the problem of determining average numbers of comparisons is open. A crucial component of our work was the development of an algorithm to compute $E[W_n]$ exactly, in time polynomial in n. Is there such an algorithm for heapsort?

Appendix 1

The following is the code for Williams' method of heap construction.

```
1.  begin
2.  A[0] ← −∞
3.  for t ← 1 to n do
4.      begin
5.      p ← t; q ← ⌊t/2⌋; a ← A[t]
6.      while A[q] > a do
7.          begin
8.          A[p] ← A[q]; p ← q; q ← ⌊p/2⌋
9.          endwhile
10.     A[p] ← a
11.     endfor
12. end
```

We estimate the efficiency of this method by counting the number of 'swaps' or 'promotions', that is, the number of executions of line 8.

Appendix 2

In this appendix we present all numerical tables of this paper.

The first table is used in estimating the error bound for ω. Next are the arrays $P(2\cdot\cdot)$ through $P(9\cdot\cdot)$. The following series of tables shows the average expected values of the levels of a heap built by Williams' algorithm, for perfect heaps of sizes 3 to 4095. Following this is a table showing the values of $I(n)$, $E[W_n]$, and $E[W_n]/n$, for n up to 32. Next is a table showing the average insertion cost along the bottom level of a heap, for up to 11 levels. (We label levels starting at zero, so a perfect heap with k levels has $2^{k+1}-1$ keys.) Finally, the last table shows the average insertion cost along the bottom level of a heap in the equi-probable case, for up to 28 levels. As usual, the cost is defined as the number of swaps involved in insertion, or the number of levels the inserted key rises.

Upper bound for α (and ω)					
j	$a_{j,11}$	b_j	$\frac{1}{2^j(1+2^j)}$	$\frac{1}{2^j+1}-2^j b_j$	$\sum_{k=1}^{j}\frac{1}{2^k+1}-2^k b_k$
1	0.000737042	0.143790058	0.166666667	0.045753217	0.045753217
2	0.001724620	0.038869985	0.050000000	0.044520060	0.090273277
3	0.003698584	0.010199395	0.013888889	0.029515951	0.119789228
4	0.007637566	0.002687947	0.003676471	0.015816382	0.135605609
5	0.015477817	0.000766655	0.000946970	0.005770084	0.141375694
6	0.031008133	0.000282152	0.000240385	-0.002673143	0.138702550
7	0.061481349	0.000161232	0.000060562	-0.012885763	0.125816788
8	0.120183174	0.000131398	0.000015199	-0.029746785	0.096070002
9	0.229364719	0.000124174	0.000003807	-0.061627770	0.034442232
10	0.419680294	0.000122490	0.000000953	-0.124453946	-0.090011714
11	0.712419883	0.000122130	0.000000238	-0.249634027	-0.339645741

Note that each of the following arrays has been multiplied by the lowest common denominator of the entries of $P(k \cdot\cdot)$.

$P(2 \cdot\cdot)$	
1	0
0	1

$3 \times P(3 \cdot\cdot)$		
3	0	0
0	1	2
0	2	1

$12 \times P(4 \cdot\cdot)$			
12	0	0	0
0	8	4	0
0	4	5	3
0	0	3	9

$10 \times P(5 \cdot\cdot)$				
10	0	0	0	0
0	8	2	0	0
0	2	3	3	2
0	0	1	3	6
0	0	4	4	2

$60 \times P(6 \cdot\cdot)$					
60	0	0	0	0	0
0	32	22	6	0	0
0	28	17	11	4	0
0	0	3	9	18	30
0	0	12	20	18	10
0	0	6	14	20	20

$105 \times P(7 \cdot\cdot)$						
105	0	0	0	0	0	0
0	40	38	21	6	0	0
0	65	25	12	3	0	0
0	0	3	9	18	30	45
0	0	12	24	29	25	15
0	0	6	15	24	30	30
0	0	21	24	25	20	15

$280 \times P(8 \cdot\cdot)$							
280	0	0	0	0	0	0	0
0	150	85	37	8	0	0	0
0	130	85	41	19	5	0	0
0	0	45	85	85	55	30	0
0	0	20	44	61	65	55	35
0	0	10	26	44	60	70	70
0	0	35	53	57	55	45	35
0	0	0	4	16	40	80	140

$252 \times P(9 \cdot\cdot)$								
252	0	0	0	0	0	0	0	0
0	166	64	23	4	0	0	0	0
0	91	77	46	24	11	3	0	0
0	0	72	82	60	29	9	0	0
0	0	12	28	42	50	50	42	28
0	0	6	16	28	40	50	56	56
0	0	21	37	44	45	42	35	28
0	0	0	2	8	20	40	70	112
0	0	0	18	42	57	58	49	28

Heapsize 3	
level	avg. exp. val.
0	0.250000000000000
1	0.625000000000000

Heapsize 7	
level	avg. exp. val.
0	0.125000000000000
1	0.342857142857143
2	0.672321428571429

Heapsize 15	
level	avg. exp. val.
0	0.062500000000000
1	0.179671717171717
2	0.383151917526918
3	0.693193611943612

Heapsize 31	
level	avg. exp. val.
0	0.031250000000000
1	0.092036756202444
2	0.205202826266408
3	0.401927398975604
4	0.703027874420290

Heapsize 63	
level	avg. exp. val.
0	0.015625000000000
1	0.046590765391610
2	0.106369076035701
3	0.217526787467712
4	0.410983835577319
5	0.707805046752974

Heapsize 127	
level	avg. exp. val.
0	0.007812500000000
1	0.023441589014780
2	0.054181970537520
3	0.113423424377795
4	0.223577187727582
5	0.415428888603783
6	0.710159837591182

Heapsize 255	
level	avg. exp. val.
0	0.003906250000000
1	0.011757757874319
2	0.027348024319593
3	0.057956420289752
4	0.116921756155955
5	0.226573708524767
6	0.417630121961365
7	0.711328907795623

Heapsize 511	
level	avg. exp. val.
0	0.001953125000000
1	0.005888172762953
2	0.013739280576294
3	0.029300398623684
4	0.059838087466778
5	0.118663456645697
6	0.228064531195549
7	0.418725232269016
8	0.711911369058983

Heapsize 1023	
level	avg. exp. val.
0	0.000976562500000
1	0.002946416557569
2	0.006886086543310
3	0.014732191558490
4	0.030276321744398
5	0.060777472225995
6	0.119532371318582
7	0.228808004732981
8	0.419271356596720
9	0.712202079702404

Heapsize 2047	
level	avg. exp. val.
0	0.000488281250000
1	0.001473791668874
2	0.003447171171386
3	0.007386778859614
4	0.015229183889442
5	0.030764209351361
6	0.061246788441340
7	0.119966326431618
8	0.229179235491644
9	0.419544046167648
10	0.712347304112488

Heapsize 4095	
level	avg. exp. val.
0	0.000244140625000
1	0.000737041788102
2	0.001724619614611
3	0.003698583603685
4	0.007637565892550
5	0.015477816549026
6	0.031008133396531
7	0.061481349099988
8	0.120183173636321
9	0.229364718604341
10	0.419680293856555
11	0.712419883270046

n	$I(n)$	$E[W_n]$	$E[W_n]/n$
1	0.0000000	0.0000000	0.0000000
2	0.5000000	0.5000000	0.2500000
3	0.3333333	0.8333334	0.2777778
4	0.9166667	1.7500000	0.4375000
5	0.6666666	2.4166665	0.4833333
6	0.7500000	3.1666665	0.5277777
7	0.5500000	3.7166665	0.5309523
8	1.2416667	4.9583330	0.6197916
9	0.9416666	5.8999996	0.6555555
10	0.9817460	6.8817458	0.6881746
11	0.7579365	7.6396823	0.6945166
12	1.1023810	8.7420635	0.7285053
13	0.8357143	9.5777779	0.7367522
14	0.8951160	10.4728937	0.7480639
15	0.6959890	11.1588831	0.7439255
16	1.4828825	12.6417656	0.7901103
17	1.1495489	13.7913141	0.8112538
18	1.1653610	14.9566755	0.8309264
19	0.9252817	15.8819571	0.8358925
20	1.2628822	17.1448402	0.8572420
21	0.9831204	18.1279602	0.8632362
22	1.0263020	19.1542625	0.8706483
23	0.8100463	19.9643097	0.8680134
24	1.3763847	21.3406944	0.8891956
25	1.0668609	22.4075546	0.8963022
26	1.0942856	23.5018406	0.9039170
27	0.8644515	24.3662930	0.9024553
28	1.2001898	25.5664825	0.9130887
29	0.9291582	26.4956398	0.9136428
30	0.9802674	27.4759064	0.9158636
31	0.7683119	28.2442188	0.9111038
32	1.6552805	29.8994999	0.9343594

Williams' method	
level	avg. ins. cost
1	0.416666666666667
2	0.720833333333333
3	0.930277014652015
4	1.067833444615132
5	1.154694091234381
6	1.207784403311318
7	1.239375461326011
8	1.257765490960568
9	1.268280503126457
10	1.274205004756312
11	1.277502849956625

Equi-probable model	
level	avg. ins. cost
1	0.416666666666667
2	0.715773809523810
3	0.922005439583565
4	1.058011754862710
5	1.144213822513107
6	1.197044446680117
7	1.228537365336889
8	1.246890932073955
9	1.257392317674570
10	1.263311592215651
11	1.266607347265580
12	1.268423804437218
13	1.269416468895913
14	1.269955059486285
15	1.270245495030628
16	1.270401285930926
17	1.270484468754062
18	1.270528704068522
19	1.270552143734820
20	1.270564524587705
21	1.270571045528150
22	1.270574471256329
23	1.270576266749685
24	1.270577205810951
25	1.270577695999073
26	1.270577951421781
27	1.270578084297521
28	1.270578153318135

Acknowledgements

This paper grew out of a problem discussed by Alan Frieze at the February 1988 Bellairs Research Institute Workshop on Random Graphs and Probabilistic Algorithms. We thank conference organizer Bruce Reed and institute director Wayne Hunt for their efforts in making the workshop a success.

The first author is also grateful to Alan Frieze, Mike Saks and Rafe Wenger for various insightful observations and motivating comments, and to Dan Arena, Magnus Halldorsson, Bob Webber, and Yi Zhu for their assistance in implementing the algorithms.

References

[BS] B. Bollobás and I. Simon, Repeated random insertion into a priority queue, *Journal of Algorithms* **6** (1985), 466-477.

[C] S. Carlsson, Average-case results on heapsort, *BIT* **27** (1987), 2-17.

[De] L. Devroye, private communication.

[Do] E.E. Doberkat, An average case analysis of Floyd's algorithm to construct heaps, *Information and Control* **61** (1984), 114-131.

[Fl] R.W. Floyd, Algorithm 245, Treesort, *CACM* **7** (1964), 701.

[Fr] A. Frieze, On the random construction of heaps, to appear in *Journal of Algorithms.*

[H] W. Hoeffding, Probability inequalities for sums of bounded random variables, *J. Amer. Statist. Assoc.* **58** (1963), 13-30.

[GM] G.H. Gonnet and J.I. Munro, Heaps on heaps, *SIAM J. Computing* **15** (1986), 964-971.

[K] D.E. Knuth, "The Art of Computer Programming, Vol. 3, Sorting and Searching", Addison-Wesley, Reading, Mass. 1973.

[M] C.J.H. McDiarmid, On the method of bounded differences, *in* J. Siemons, ed., "Surveys in Combinatorics, 1989", London Math. Soc. Lecture Note Series 141, Cambridge University Press 1989.

[MR] C.J.H. McDiarmid and B. Reed, Building heaps fast, *Journal of Algorithms* **10** (1989), 352-365.

[PS] T. Porter and I. Simon, Random insertion into a priority queue structure, *IEEE Transactions on Software Engineering* **SE-1** (Sept. 1975), 292-298.

[W] J.W.J. Williams, Algorithm 232, *CACM* **7** (1964), 347-348.