Figure 1: Graph of S_k for a random walk of 10^6 steps.

[14] M. Kolountzakis, message posted to TheoryNet on Jan. 4, 1990.

[15] National Bureau of Standards, *Handbook of Mathematical Functions*, M. Abramowitz and I. A. Stegun, eds., U.S. Gov. Printing Office, 9th printing, 1970.

[16] A. M. Odlyzko, Extremal and statistical properties of trigonometric polynomials with ± 1 and $0, 1$ coefficients, manuscript in preparation.

[17] P. Révész, *Random Walk in Random and Nonrandom Environments*, World Scientific, 1990.

[18] M. R. Schroeder, *Number Theory in Science and Communication*, Springer, 1984.

[19] F. Spitzer, *Principles of Random Walk*, Van Nostrand, 1964.

[20] J. vom Scheidt, Random functions in physical and mechanical systems, pp. 55–64 in *Proc. Fifth European Conference on Mathematics in Industry*, M. Heiliö, ed., Kluwer, 1991.

[21] S. Yakowitz and E. Lugosi, Random search in the presence of noise, with application to machine learning, *SIAM J. Sci. Stat. Comput. 11* (1990), 702–712.

[22] A. A. Zhigljavsky, *Theory of Global Random Search*, Kluwer, 1991.

References

[1] I. Althöfer and K.-U. Koschnick, On the deterministic complexity of searching local maxima, *Discrete Appl. Math. 43* (1993), 111–113.

[2] M. Avriel and D. J. Wilde, Optimal search for a maximum with a sequence of simultaneous function evaluations, *Management Science 12* (1966), 722–731.

[3] M. N. Barber and B. W. Ninham, *Random and Restricted Walks*, Gordon and Breach, 1970.

[4] S. Ben-David, Can finite samples detect singularities of real-valued functions, pp. 390–399 in *Proc. 24th ACM Symp. Theory Comp.*, ACM, 1992.

[5] K.-L. Chung, W.-C. Chen, and F.-C. Lin, On the complexity of search algorithms, *IEEE Trans. Computers 41* (1992), 1172–1176.

[6] R. Cole and K. C. Yap, Shape from probing, *J. Algorithms 8* (1987), 19–38.

[7] W. Feller, *An Introduction to Probability Theory and Its Applications*, vol. 1, 3rd ed., Wiley, 1968.

[8] A. S. Goldstein and E. M. Reingold, A Fibonacci version of Kraft's inequality applied to discrete unimodal search, *SIAM J. Comput. 22* (1993), 751–777.

[9] B. Hajek, Locating the maximum of a simple random sequence by sequential search, *IEEE Trans. Information Theory IT-33* (1987), 877–881.

[10] M.-Y. Kao, J. H. Reif, and S. R. Tate, Searching in an unknown environment: An optimal randomized algorithm for the cow-path problem, pp. 441–447 in *Proc. 4th ACM-SIAM Symp. Discrete Math.*, 1993.

[11] R. M. Karp and W. L. Miranker, Parallel minimax search for a maximum, *J. Comb. Theory 4* (1968), 19–35.

[12] J. Kiefer, Sequential minimax search for a maximum, *Proc. Amer. Math. Soc. 4* (1953), 502–506.

[13] D. E. Knuth, *The Art of Computer Programming Vol. 3: Sorting and Searching*, Addison–Wesley, Reading, 1973.

weight that is approximately $1/r$ (at least for $r > \log n$, say), and so the average weight assigned to m (and also all its neighbors) is about

$$S_m = \sum_{r=1}^{\infty} r^{-1} Pr(M_m - S_m = r - 1) \ . \tag{6.3}$$

However, by Lemma 5,

$$\begin{aligned} S_m &\sim \left(\frac{2}{\pi m}\right)^{1/2} \int_0^{\infty} (u+1)^{-1} \exp(-u^2/(2m)) du \\ &\sim (2\pi m)^{-1/2} \log m \quad \text{as} \quad m \to \infty \ . \end{aligned} \tag{6.4}$$

Hence the total weight assigned to all the points $1, 2, \ldots, n$, averaged over all the walks in W, is

$$\sim \sum_{m=1}^{n} S_m \sim (2\pi^{-1} n)^{1/2} \log n \quad \text{as} \quad n \to \infty \ . \tag{6.5}$$

We note that even on a local scale this linear search algorithm on average loses a factor of 2 compared to algorithms that can backtrack to the position of the previous probe. This occurs since it can go forward only by $M_m - S_m + 1$, and not by almost $2(M_n - S_m)$, which is how far it can go on average and still obtain a valid bound. What is most significant, though, is that this algorithm is slower by a factor of $\log n$ than the algorithm of Theorem 2, which uses global information.

Simulations suggest that the ratio of the average running time of Algorithm K to the asymptotic value in (1.5) converges to 1 slowly, approximately at the rate $1 + O((\log n)^{-1})$. The standard deviation of the running time of Algorithm K seems to be not much smaller than the mean.

$\delta, p > 0$ requires $O(1)$ probes as $n \to \infty$. All that is necessary is to probe S_k at $k = \lfloor \epsilon n \rfloor$, $\lfloor 2\epsilon n \rfloor, \ldots$, for a small $\epsilon > 0$ and select the maximum of these S_k as the estimate of M_n. The probability that $|S_{k+j} - S_k| > \delta n^{1/2}$ for a fixed k and $0 \leq j \leq \epsilon n$ is $< \exp(-c'\delta^2/\epsilon)$ for a constant $c' > 0$ (independent of δ, s, and n). Hence the estimate we obtain will be off by more than $\delta n^{1/2}$ with probability $O(\epsilon^{-1} \exp(-c'\delta^2/\epsilon))$ as $n \to \infty$. Therefore for ϵ sufficiently small, this probability of error will be $< p$.

The number of probes, $\lfloor 1/\epsilon \rfloor$, in the scheme outlined above is on the order of $-\delta^{-2} \log p$. With more effort (using several stages, as in Sections 3 and 4) this number can be lowered substantially.

To obtain the exact complexity of the optimal algorithm for the full range of values of δ and p is likely to be hard. We just note that one can estimate M_n to within $\delta n^{1/2}$ with error probability $O(n^{-10})$ (or any inverse polynomial bound) at a cost of $O(\log n)$. This follows easily from the estimate for the probability of W (see (3.16)). If a walk is in W, then probing at a sequence of points $k = K, 2K, \ldots$, where $K = \lfloor \epsilon n (\log n)^{-1} \rfloor$ will yield an estimate that is guaranteed to be within $\delta n^{1/2}$. Using a more elaborate probing strategy, one can again lower the $O(\log n)$ cost.

6. Linear search

In this section we sketch a proof of Theorem 4. Since the methods are almost the same as those used in proving Theorem 2, but simpler, we do not present full details.

Lemma 4. *For any integer $q \geq 0$,*

$$Pr(M_n - S_n = q) = \sum_{\substack{k \\ k \equiv n (\bmod\ 2) \\ k \geq -q}} 2 \frac{k + 2q + 1}{n + k + q + 2} \binom{n}{\frac{n+k+2q}{2}} 2^{-n} \ . \tag{6.1}$$

Proof. This result follows immediately from Lemma 2. ■

Lemma 5. *We have uniformly in q, $0 \leq q \leq n^{3/5}$, that*

$$Pr(M_n - S_n = q) \sim \left(\frac{2}{\pi n} \right)^{1/2} \exp(-q^2/(2n)) \quad as \quad n \to \infty \ . \tag{6.2}$$

Proof. This result follows from Lemmas 1 and 4. ■

We now prove Theorem 4. If the searcher's most recent question was about S_m, then she knows M_m, and therefore her next question will be about S_r, where $r = m + M_m - S_m + 1$. Therefore, if the walk is in the class W (defined in Section 3), than the point m gets assigned

To make the above argument rigorous, it is only necessary to prove quantitative estimates of how much $S_m - S_{m+k}$ varies when $M_n - S_m$ is small. This is not difficult, but somewhat tedious, so we do not do it here.

5. Approximate maxima

Suppose we wish to determine M_n to within an additive factor of $h \geq 1$. The worst case running time is then roughly proportional to n/h. If we probe at $2h, 4h, 6h, \ldots$, the maximal value we obtain will be within h of M_n. On the other hand, if there were some strategy that always determined M_n to within h by using fewer than $\sim n(2h+4)^{-1}$ probes as $n \to \infty$, then this strategy would have some intervals of $> 2h+2$ points entirely unprobed, even when faced with an adversary that gave answers consistent with the random walk $S_{2k} = 0$, $S_{2k+1} = 1$. Hence this strategy would not prove that $M_n < h+2$, which is a contradiction. With a bit more effort one can obtain better bounds.

The worst case cost of determining M_n to within h grows like n/h, and so large savings are available for big h, compared to the cost of determining M_n exactly. On the other hand, the average cost of determining M_n to within h for small h is still $\sim c_0 n^{1/2}$ as $n \to \infty$. This is clearly an upper bound, since by determining M_n exactly we determine it to within h. To see that this is a lower bound, we examine the proof of the lower bound of Theorem 2. It can be seen there that almost all the probes in a minimal proof of M_n are in regions where $M_n - S_m$ is on the order of $n^{1/2}$. Therefore the freedom to lengthen the interval between consecutive probes by an additive factor of about $2h$ is of negligible importance for h small, $h = o(n^{1/2})$. We conclude therefore that to determine M_n to within h for any $h = o(n^{1/2})$ still costs $\geq (c_0 + o(1))n^{1/2}$ as $n \to \infty$ (on average).

Suppose we wish to determine M_n to within an additive factor of $\delta n^{1/2}$ for some $\delta > 0$. The worst case and average case costs are still on the order of $n^{1/2}$ if we insist on obtaining the correct answer. What happens, though, if we allow a nonzero probability p of error? If

$$p > \operatorname{erfc}(\delta 2^{1/2}) = 2\pi^{-1/2} \int_{\delta 2^{1/2}}^{\infty} \exp(-t^2) dt \ , \tag{5.1}$$

then the cost is 0, for large n. The reason is that $\operatorname{erfc}(\delta 2^{1/2})$ is asymptotic to the probability that $M_n > 2\delta n^{1/2}$. Hence the searcher can declare $\delta n^{1/2}$ to be her estimate of M_n, and she will be wrong by more than $\delta n^{1/2}$ with probability $\lesssim p$.

More generally, to determine M_n to within $\delta n^{1/2}$ with error probability $\leq p$ for any fixed

However, by Corollary 1, the expected number of k that satisfy (4.11) is

$$\leq c_{14} 2^{N-r} \log n \ . \tag{4.12}$$

Therefore the expected value of b_r is $\leq c_{14} \log n$, and so the expected cost of one algorithm is $O((\log n)^2)$, which proves Theorem 3.

It is easy to prove a lower bound of the form $c_{15} \log n$ for the average cost of any algorithm that determines M_n correctly with probability $\geq 1 - n^{-10}$. If there were an algorithm that did this at average cost $\leq \epsilon \log n$, then with probability $\geq 1/2$, there would be a gap between consecutive positions that are probed that was $> n(2\epsilon \log n)^{-1}$. Lemma 3 implies that $\max |S_k| < 2n^{1/2}$ with probability $> 9/10$, so we would have that the probability that both $\max |S_k| < 2n^{1/2}$ and that there is an interval that is not probed of length $> n(2\epsilon \log n)^{-1}$ is $\geq 2/5$. But the probability that $|S_{m+k} - S_m| > 10n^{1/2}$ for m and $m + k$ both inside that interval can be easily shown to be

$$> \exp(-c_{16} \epsilon \log n) \ , \tag{4.13}$$

which contradicts our basic assumption for ϵ small enough. This proves the $c_{15} \log n$ lower bound.

We now sketch how to prove a lower bound of the form $c_{17}(\log n)^2$. This would be a bound for any proof. The idea is similar to that of the proof of Theorem 2. Consider probes near m. Typically, $|S_m - S_{m+k}|$ will be on the order of $k^{1/2}$. How far can consecutive probes be situated so as to ensure that a mistake is not made with probability $> n^{-10}$? If we probe S_m and S_{m+k}, where $k = \epsilon(M_n - S_m)^2$, then the probability of an excursion in that interval that exceeds $M_n - S_m$ is roughly

$$\exp(-c_{18} \epsilon^{-1}) \ . \tag{4.14}$$

Therefore we must have $\epsilon < c_{19}(\log n)^{-1}$. Hence we can expect that a minimal proof will have expected cost

$$\geq c_{20} \sum_{q=1}^{n^{1/3}} \frac{\log n}{q^2} E(D_q) \ . \tag{4.15}$$

Since Corollary 1 can be shown to be sharp for $q \leq n^{1/3}$, this gives expected cost of any proof of

$$\geq c_{21} \sum_{q=1}^{n^{1/3}} q^{-1} \log n \geq c_{22} (\log n)^2 \ . \tag{4.16}$$

each j such that $j2^{N-r}$ is in one of the b_r blocks. This gives us $2b_r$ blocks of length 2^{N-r-1} each. Suppose these blocks are

$$B'_h = \{j_h 2^{N-r-1} + i : \quad 0 \leq i < 2^{N-r-1}\} \ , \quad h = 1, \ldots, 2b_r \ . \tag{4.2}$$

Let

$$M'_n = \max\{S_k : \ k = j_h 2^{N-r-1}, \ \ 1 \leq h \leq 2b_r\} \ . \tag{4.3}$$

We select B'_h to be among the b_{r+1} blocks of stage $r+1$ if

$$M'_n - S_{j_h 2^{N-r-1}} \leq 2c_6 2^{(N-r-1)/2} (\log n)^{1/2} \ . \tag{4.4}$$

Note that if B'_h satisfies (4.4), then

$$M_n - S_k \leq (3 + 2^{1/2}) c_6 2^{(N-r-1)/2} (\log n)^{1/2} \tag{4.5}$$

holds for every $k \in B'_h$. If any k satisfies

$$0 \leq M_n - S_k \leq c_6 2^{(N-r-1)/2} (\log n)^{1/2} \ , \tag{4.6}$$

then it satisfies (4.1), and so is in one of the b_r blocks B of stage r, and therefore in one of the b_{r+1} blocks B'_h of stage $r+1$.

The algorithm terminates at stage R, where R is the smallest integer such that

$$c_6 2^{(N-R)/2} (\log n)^{1/2} \geq 2^{N-R} \ . \tag{4.7}$$

At that point we probe every point k in any of the b_R blocks of stage R, which costs

$$\leq b_r 2^{N-R} = O(b_R \log n) \ . \tag{4.8}$$

The total cost of the algorithm is

$$\sum_{r=0}^{R-1} b_r + O(b_R \log n) \ . \tag{4.9}$$

(If we run into a walk that is provably not in W, we probe every k, and this costs $O(n^{-9})$ on average.) We now need to bound the expected values of b_r. By (4.5) (applied with $r-1$ instead of r) we see that there are

$$\geq b_r 2^{N-r} \tag{4.10}$$

values of k such that

$$M_n - S_k \leq 5c_6 2^{(N-r)/2} (\log n)^{1/2} \ . \tag{4.11}$$

where

$$c_{13} = \int_0^\infty dy \int_0^1 \frac{\exp(-y^2/(2w))dw}{w^{1/2}(1-w)^{1/2}} \int_0^1 \exp(-y^2x^2/(2-2w))dx \ . \tag{3.53}$$

The integral on x in (3.53) can be expressed in terms of the erf function [15], but it is not clear whether there is a simple closed form expression for c_{13} (and consequently for the constant c_0 of Theorem 1).

The sum

$$U_B = \frac{1}{2} \sum_{q=0}^{N} \frac{1}{q+1} \sum_{m=8N}^{n-8N} B(n,m,q) \tag{3.54}$$

is estimated similarly, and we find that

$$U_B \sim 2\pi^{-1} J_n \quad \text{as} \quad n \to \infty \ , \tag{3.55}$$

where

$$J_n = \int_0^\infty \frac{dq}{q+1} \int_0^n \frac{du}{u^{1/2}(n-u)^{1/2}} \int_0^{q/2} \exp(-2v^2/u - q^2/(2n-2u))dv \ . \tag{3.56}$$

The changes of variables $v = qx/2$, $u = nw$, and $q = n^{1/2}y$ show that $J_n = I_n/2$. Collecting all our estimates yields the claim of the Proposition. ■

4. Exact maxima with nonzero error probabilities

In this section we consider the problem of computing the exact value of M_n, but this time we allow the answer to be wrong with probability $\leq n^{-10}$. The algorithm will produce a value that equals M_n whenever the walk is in the class W defined in Section 3. (We can replace n^{-10} by $n^{-\alpha}$ for any constant α, but the estimates derived in Section 2 make it convenient not to choose α too large.)

For simplicity of exposition, we will assume that $n = 2^N - 1$ for some integer $N \geq 1$. Modifications required for other n are minor. The algorithm of Theorem 3 consists of $\leq N$ stages. In stage r, $0 \leq r < N$, there will be b_r disjoint blocks of 2^{N-r} consecutive integers each such that if B is one of those blocks, $B = \{j2^{N-r} + i : 0 \leq i < 2^{N-r}\}$ for some j. Further, if the walk is in W, and

$$0 \leq M_n - S_k \leq c_6 2^{(N-r)/2} (\log n)^{1/2} \tag{4.1}$$

holds for some k, where c_6 is the constant of (3.15) that defines W, then k is in one of the b_r blocks B. We will also know S_k for the smallest k $(= j2^{N-r})$ in the block B. Initially we start with $r = 0$, $b_0 = 1$. To go from stage r to stage $r+1$, we probe S_k for $k = j2^{N-r} + 2^{N-r-1}$ for

Similar bounds apply to the sum of $B(n,m,q)$ over $0 \leq m \leq 8N$ and the corresponding sums of $A(n,m,q)$ and $B(n,m,q)$ over $n-8N \leq m \leq n$. We conclude that

$$V_n = \frac{1}{2}\sum_{q=0}^{N} \frac{1}{q+1} \sum_{m=8N}^{n-8N} (A(n,m,q)+B(n,m,q)) + O(n^{1/4}(\log n)) \ . \tag{3.44}$$

In the range $0 \leq q \leq N$, $8N \leq m \leq n-8N$, we apply Lemma 1 to the individual terms in the definitions (3.6) and(3.7) of $A(n,m,q)$ and $B(n,m,q)$. We find that

$$\begin{aligned} U_A &= \frac{1}{2}\sum_{q=0}^{N} \frac{1}{q+1} \sum_{m=8N}^{n-8N} A(n,m,q) \\ &\sim \pi^{-1} \sum_{q=0}^{N} \frac{1}{q+1} \sum_{m=8N}^{n-8N} m^{-1/2}(n-m)^{-1/2} \sum_{j=0}^{q} \exp(-q^2/(2m) - j^2/(2n-2m)) \end{aligned} \tag{3.45}$$

as $n \to \infty$. The Euler-Maclaurin summation formula next shows that

$$U_A \sim \pi^{-1} I_n \quad \text{as} \quad n \to \infty \ , \tag{3.46}$$

where

$$I_n = \int_0^\infty \frac{dq}{q+1} \int_0^n \frac{du}{u^{1/2}(n-u)^{1/2}} \int_0^q \exp(-q^2/(2u) - v^2/(2n-2u)) dv \ . \tag{3.47}$$

To simplify the expression for I_n, we first substitute $v = qx$, then $u = nw$, and finally $q = n^{1/2}y$. We find that

$$I_n = n^{1/2} \int_0^\infty \left(1 - \frac{1}{n^{1/2}y+1}\right) dy \int_0^1 \frac{\exp(-y^2/(2w))dw}{w^{1/2}(1-w)^{1/2}} \int_0^1 \exp(-y^2x^2/(2-2w))dx \ . \tag{3.48}$$

Since

$$\int_0^1 \frac{dw}{w^{1/2}(1-w)^{1/2}} = \pi \tag{3.49}$$

and

$$\exp\left(-\frac{y^2}{2w}\right) \leq \exp(-y^2/2) \quad \text{for} \quad 0 \leq w \leq 1 \ , \tag{3.50}$$

we have

$$\begin{aligned} &\int_0^\infty \frac{dy}{n^{1/2}y+1} \int_0^1 \frac{\exp(-y^2/(2w))dw}{w^{1/2}(1-w)^{1/2}} \int_0^1 \exp(-y^2x^2/(2-2w))dx \\ &\qquad \leq \pi \int_0^\infty \frac{\exp(-y^2/2)dy}{n^{1/2}y+1} = O(n^{-1/2}\log n) \ . \end{aligned} \tag{3.51}$$

Therefore

$$I_n = c_{13} n^{1/2} + O(\log n) \ , \tag{3.52}$$

Proof. By the definition (3.10) of D_q and (3.18) of V_n, we have

$$V_n = \frac{1}{2}\sum_{q=0}^{\infty} \frac{1}{q+1} E(D_q) \ . \tag{3.37}$$

By Proposition 1 and Eq. (3.11),

$$V_n = \frac{1}{2}\sum_{q=0}^{\infty} \frac{1}{q+1} \sum_{m=0}^{n} (A(n,m,q) + B(n,m,q)) \ . \tag{3.38}$$

On the right-hand side of Eq. (3.38), all terms with $q > n$ vanish. Further, if $q > N$, where again N is given by (3.19), then for any m,

$$2^{-m}\binom{m}{\left\lfloor \frac{m+q+1}{2} \right\rfloor} \ < \ n^{-20} \ , \tag{3.39}$$

$$2^{-m}\binom{n-m}{\left\lfloor \frac{m-n-q}{2} \right\rfloor} \ < \ n^{-20} \ , \tag{3.40}$$

and so

$$V_n = \frac{1}{2}\sum_{q=0}^{N} \frac{1}{q+1} \sum_{m=0}^{n} (A(n,m,q) + B(n,m,q)) + O(n^{-10}) \ . \tag{3.41}$$

We next show that the terms in the sum in (3.41) with either very small or very large m are negligible. We can deduce from Lemma 1 that

$$\binom{r}{\lfloor \frac{r}{2} \rfloor + k} \le c_8 2^r (r+1)^{-1/2} \exp(-\epsilon k^2/(r+1)) \tag{3.42}$$

holds for all $r \ge 0$ and all integers k, for some positive constants c_8 and ϵ. Hence for $\delta = \epsilon/5$, and n large,

$$\begin{aligned}
&\sum_{q=0}^{N} \frac{1}{q+1} \sum_{m=0}^{8N} A(n,m,q) \\
&\le c_8^2 \sum_{q=0}^{N} \frac{1}{q+1} \sum_{m=0}^{8N} (m+1)^{-1/2}(n-m+1)^{-1/2} \sum_{j=0}^{q} \exp(-\delta q^2/(m+1) - \delta j^2/n) \\
&\le c_9 n^{-1/2} \sum_{q=0}^{N} \frac{1}{q+1} \sum_{m=0}^{8N} \sum_{j=0}^{q} \exp(-\delta q^2/(m+1) - \delta j^2/n) \\
&\le c_{10} n^{-1/2} \sum_{q=0}^{N} \frac{1}{q+1} \sum_{m=0}^{8N} n^{1/2} \exp(-\delta q^2/(m+1)) \\
&= c_{10} \sum_{m=0}^{8N} \sum_{q=0}^{N} \frac{1}{q+1} \exp(-\delta q^2/(m+1)) \\
&\le c_{11} \sum_{m=0}^{8N} (m+1)^{-1/2} \le c_{12} n^{1/4} (\log n)^{1/2} \ .
\end{aligned} \tag{3.43}$$

by Corollary 1. Thus this case also contributes a negligible amount to the running time of the algorithm. When (3.27) is not satisfied, but the walk is in W, the step size k defined by (3.28) satisfies

$$k = 2(M_n - S_m)(1 + O(n^{-1/10})) \ . \tag{3.31}$$

We now define the concept of a weight of a point. If m and $m+k$ are the two successive points that are probed, and $m \leq h < m+k$, we say that $wt(h)$, the weight of h, is $1/k$. The total weight $wt(w)$ of a walk w is

$$wt(w) = \sum_{h=0}^{n-1} wt(h) \ . \tag{3.32}$$

We see that $wt(w)$ is just the number of probes of w. For $w \in W$, (3.31) and the condition (3.15) imply that $wt(h) \leq 1$ if $M_n - S_h \leq 3n^{1/6}$, and

$$wt(h) = (2(M_n - S_h + 1))^{-1}(1 + O(n^{-1/10})) \tag{3.33}$$

otherwise. Hence

$$wt(w) = (1 + O(n^{-1/10})) \sum_{h=0}^{n-1} (2(M_n - S_h + 1))^{-1} + O(n^{1/3}) \tag{3.34}$$

for a walk $w \in W$. This proves that the average running time of the algorithm is $\leq (1 + o(1))V_n$ as $n \to \infty$.

To prove the lower bound of Theorem 2, we again use the concept of a weight of a point. Suppose the pairs $(0,0), (m_1, S_{m_1}), \ldots, (m_r, S_{m_r})$ form a minimal proof of M_n. An integer h, $0 \leq h < n$, is assigned weight $wt(h) = (m_{i+1} - m_i)^{-1}$ if $m_i \leq h < m_{i+1}$, $0 \leq i \leq r$. Then

$$r = \sum_{h=0}^{n-1} wt(h) \ . \tag{3.35}$$

The same analysis was used for the upper bound proof of Theorem 2 shows that for walks $w \in W$, Eq. (3.34) holds. This shows that the average cost of a minimal proof is $\geq (1 + o(1))V_n$ as $n \to \infty$.

To conclude the proof of Theorem 2, we need to estimate V_n.

Proposition 2. *We have*

$$V_n \sim c_0 n^{1/2} \quad as \quad n \to \infty \ , \tag{3.36}$$

where c_0 is given by (1.2).

$$
\begin{aligned}
&\le \frac{2}{N}\sum_{w} |\{k : 0 \le k \le n, \quad (3.24) \text{ holds}\}| \\
&\le \frac{2}{N} 2^n \cdot \sum_{q=0}^{Q} E(D_q) \\
&\le 2^n c_7 N^{-1} Q^2 = O(2^n \log n) \ . \qquad (3.25)
\end{aligned}
$$

Since W satisfies (3.16), the expected number of r, $0 \le r \le n/N$, that satisfy (3.21) is $O(\log n)$. (Walks outside W contribute $O(n \cdot n^{-10}) = O(n^{-9})$.) Therefore the expected cost of the second phase is $O(n^{1/4}(\log n)^2)$.

Suppose now that the second phase of the algorithm is complete. The third and final phase is to scan the walk from left to right. At any given time $T = 0, 1, 2, \ldots$, we will have an integer m such that we will know S_m, and an estimate $M^{\#}$ of M_n that will satisfy

$$M_n - c_6 n^{1/8} (\log n)^{1/2} \le M^{\#} \le M_n \ . \qquad (3.26)$$

We will have proved rigorously that $M_m \le M^{\#}$. At time $T = 0$ we start with $m = 0$, $M^{\#} = M^*$. Suppose we are at time T. If

$$M^{\#} - S_m \le n^{1/6} \ , \qquad (3.27)$$

then we probe S_{m+1}. If $S_{m+1} > M^{\#}$ (which means $S_{m+1} = M^{\#} + 1$), we increase $M^{\#}$ by 1. If (3.27) is not satisfied, we probe S_{m+k}, where

$$k = 2(M^{\#} - S_m) - 10 c_6 (M^{\#} - S_m)^{1/2} (\log n)^{1/2} \ . \qquad (3.28)$$

(If $m + k > n$, we probe n, obviously.) If $S_{m+k} - S_m$ does not satisfy (3.15), then our walk is not in W, and we abort this approach and probe every position. If the walk is in W, though, then (3.15) is satisfied, and since $M^{\#} - S_m > n^{1/6}$, we obtain a rigorous proof that $S_h < M^{\#}$ for $m \le h \le m + k$. Therefore at the next time $T + 1$ we use the same value $M^{\#}$ but replace m by $m + k$.

The algorithm described above clearly finds the exact value of M_n. What is its average cost? The average cost of the first two phases has already been shown to be $O(n^{1/2}/(\log n))$. In the third phase, how often do we encounter condition (3.27)? If (3.27) is satisfied, then

$$M_n - S_m \le 2n^{1/6} \ , \qquad (3.29)$$

since (3.23) is valid. Therefore the expected number of times (3.27) occurs is

$$\le \sum_{q=0}^{\lfloor 2n^{1/6} \rfloor} E(D_q) = O(n^{1/3}) \qquad (3.30)$$

The cost of this phase is $\sim n^{1/2}(\log n)^{-1}$, and is the same for all walks. If during this or any of the other phases we find any m and k that violate (3.15), we conclude that the walk we are examining is not in W, and so we probe all positions $1, 2, \ldots, n$. Since this has probability $\leq n^{-10}$ of occurring, it contributes a negligible amount to the cost of the algorithm. The description of the algorithm that follows assumes we do not find any violations of (3.15).

Next we let

$$M' = \max(S_0, S_N, S_{2N}, \ldots) \ . \tag{3.20}$$

Then $M' \leq M_n$, and if the walk is in W, then $M_n \leq M' + c_6 n^{1/4} \log n$. That estimate is not adequate for our purposes, and so we use a second phase. If for some $r = 0, 1, \ldots$, we have

$$S_{rN} > M' - c_6 n^{1/4} \log n \ , \tag{3.21}$$

then we probe

$$S_{rN \pm jm}, \qquad j = 1, 2, \ldots, \lceil N/m \rceil \ , \tag{3.22}$$

where $m = \lfloor n^{1/4} \rfloor$. Note that if we are in W, then any k with $S_k = M_n$ must satisfy $|k - rN| < N$ for some r for which (3.21) holds. Since the second phase probes at intervals of $\leq n^{1/4}$, if we let M^* be the maximum of all the S_k found in either the first or the second phase, then

$$M_n - c_6 n^{1/8} (\log n)^{1/2} \leq M^* \leq M_n \tag{3.23}$$

holds, provided our walk is in W.

We next estimate the cost of the second phase. To do this we need to know the average number of points rN, $0 \leq r \leq n/N$, that satisfy (3.21). If a walk is in W and (3.21) holds, then

$$S_k > M' - 2c_6 n^{1/4} \log n$$

for all k with $|k - rN| \leq N$, and therefore also

$$S_k > M_n - 3c_6 n^{1/4} \log n \ . \tag{3.24}$$

Let w denote a walk of n steps, and $F(w)$ the number of r, $0 \leq r \leq n/N$, that satisfy (3.21). To each r counted by $F(w)$, we can associate $\lfloor N/2 \rfloor$ values of k such that $|k - rN| \leq N/2$ and such that (3.24) holds. These sets of $\lfloor N/2 \rfloor$ values of k associated to different r are disjoint. Therefore for $Q = \lfloor 3c_6 n^{1/4} \log n \rfloor$, we have

$$\sum_{w \in W} F(w) \ \leq \ \sum_{w \in W} \frac{2}{N} |\{k : \ 0 \leq k \leq n, \ \ (3.24) \text{ holds}\}|$$

will next probe S_{m+r}, where r is a little less than $2(B - S_m)$. If she is dealing with a walk in W, S_{m+r} will be close to S_m, and this will establish (whether the walk is in W or not) that $S_k \leq B$ for $m \leq k \leq m + r$. To make this procedure efficient, it is necessary to obtain a good estimate of M_n. This is done in two preliminary stages, which are guaranteed to determine M_n to within a small error if the walk is indeed in W.

The running time of the algorithm can be estimated (heuristically, at least) from the discussion above. For a walk with $S_m = 500$, $S_{m+h} = 550$, $h = 3 \cdot 10^4$, $M_n \geq 1500$, we will need $\leq h/(2 \cdot (1500 - 650)) = h/1700$ probes to establish $S_k \leq 1500$ for $m \leq k \leq m + h$, provided each probe yields a value $S_k \leq 650$. In general, if we have a good estimate of M_n and are dealing with a random walk in W, we need one probe of some S_r with r near k for every $2(M_n - S_k)$ positions. Therefore we expect a total of about

$$\sum_{k=1}^{n} \frac{1}{2(M_n - S_k + 1)} \tag{3.17}$$

probes for a walk in W. Since walks not in W are rare, we expect the algorithm to have average running time of $(1 + o(1))V_n$, where

$$V_n = E\left(\sum_{k=1}^{n} \frac{1}{2(M_n - S_k + 1)} \right) . \tag{3.18}$$

That is what we will prove. At the end of this section we will provide an asymptotic estimate of V_n.

Before we present the precise description of the algorithm and a rigorous analysis of its running time, we explain how the lower bound of Theorem 2 is obtained. Consider again the case where we know that $h = 3 \cdot 10^4$, $S_m = 500$, $S_{m+h} = 550$, but this time assume further that we know exactly that $M_n = 1510$ and that $400 \leq S_k \leq 650$ for $m \leq k \leq m+h$. In that case to prove that $S_k \leq 1510$ for $m \leq k \leq m + h$, we need to provide $\geq (h - 2220)/(2(1510 - 400)) = (h - 2220)/2220$ values of S_k (and possibly more, if many of the values of S_k are larger than 400). In general, for a walk in W, near k we need to provide a value approximately every $2(M_n - S_k)$ positions. Therefore for a walk in W the number of values of k for which S_k has to be revealed is close to the quantity in (3.17). Since walks outside W have probability $\leq n^{-10}$, the number of values of S_k that have to be revealed is likely to be $\geq (1 + o(1))V_n$. A rigorous proof of this follows from the analysis of the proof of the upper bound of Theorem 2.

We now describe the algorithm. The first phase consists of probing S_N, S_{2N}, $S_{3N}, \ldots$, where

$$N = \lfloor n^{1/2} \log n \rfloor . \tag{3.19}$$

■

It is easy to obtain sharper results than those of Corollary 1. For example, it can be shown that $E(D_q) \le (1 + o(1))(4q + 2)$ as $n \to \infty$, uniformly in q. (A similar result is presented as Theorem 13.22 in [17], but it is proved there only for individual high excursions of the infinite random walk.)

We now proceed to the heart of the proof of Theorem 2. We define $W = W(n)$ to be the set of random walks of n steps such that

$$|S_{m+k} - S_m| \le c_6 k^{1/2} (\log n)^{1/2} \tag{3.15}$$

for all $k, m \ge 0$ with $k + m \le n$. If we choose c_6 large enough, then Lemma 1 shows that condition (3.15) fails for any fixed k and m with probability $\le n^{-12}$, and so

$$Pr(W) \ge 1 - n^{-10} \ . \tag{3.16}$$

Both the upper and lower bounds of Theorem 2 are based on the observation that walks not in W do not appreciably affect the running times of algorithms, since they are rare, while walks in W are well behaved and are easy to analyze. We demonstrate with a simple example. Suppose that $m = 10^5$, $h = 3 \cdot 10^4$, and the searcher knows that $S_m = 500$, $S_{m+h} = 550$, and that $1500 \le M_n$. If the searcher thinks that the walk is in W, and (3.15) gives her $S_k \le 650$ for $m \le k \le m + h$, then she can probe S_{m+1850} as the next step. If it turns out that $S_{m+1850} \le 650$, as expected, then there will be no need to probe S_k for any k with $m < k < m + 1850$, as all such k will have to satisfy $S_k \le 1500$. Of course, it might turn out that the searcher's assumption that the walk is in W is wrong, and she might find that $S_{m+1850} = 702$, for example. In that case the strategy will be to probe each of $S_1, \ldots, S_n$. Since the cost of a complete search is n probes, and such searches will only need to be done for walks not in W, which have probability $\le n^{-10}$, their contribution to the expected cost of the algorithm will be $\le n^{-9}$, which is negligible. Note that the searcher does not obtain a proof that the walk she is investigating is in W. For example, if $S_{m+1850} = 650$, it might happen that $S_{m+1000} = 1500$, which proves the walk is not in W, but this is not discovered by the searcher, since she never probes S_{m+1000}. The point is that the algorithm does prove rigorously that $S_k \le 1500$ for $m \le k \le m + 1850$, and that the additional probes of S_k for $m < k < m + 1850$ arise only for walks not in W, which are rare. In general, if at a certain stage of the algorithm the searcher knows that $M_n \ge B$, and she knows the value of S_m, she

Proof. We decompose the n-step random walk into two random walks, consisting of the initial m and final $n-m$ steps, respectively. Since these two walks are independent, we have, for any integers h and k such that $h \equiv m (\mathrm{mod}\ 2)$, $k \geq 0$, that

$$\begin{aligned} Pr(M_n = k \ \text{ and } \ S_m = h) \ &= \ Pr(M_m = k \ \text{ and } \ S_m = h) \cdot Pr(M_{n-m} \leq k-h) \\ &\qquad + Pr(M_m < k \ \text{ and } \ S_m = h) \cdot Pr(M_{n-m} = k-h) \ . \end{aligned} \tag{3.8}$$

We now substitute the formulas from Lemmas 2 and 3 and sum on those h and k that satisfy $h = k - q$, $h \equiv m (\mathrm{mod}\ 2)$, $k \geq 0$. For example,

$$\begin{aligned} &\sum_{\substack{h \\ h \equiv m (\mathrm{mod}\ 2) \\ h \geq -q}} Pr(M_m < h + q \ \text{ and } \ S_m = h) \cdot Pr(M_{n-m} = q) \\ &= Pr(M_{n-m} = q) \sum_{\substack{h \\ h \equiv m (\mathrm{mod}\ 2) \\ h \geq -q}} \{Pr(S_m = h) - Pr(M_m \geq h + q \ \text{ and } \ S_m = h)\} \\ &= 2^{-n} \binom{n-m}{\lfloor \frac{n-m-q}{2} \rfloor} \sum_{\substack{h \\ h \equiv m (\mathrm{mod}\ 2) \\ h \geq -q}} \left\{ \binom{m}{\frac{m+h}{2}} - \binom{m}{\frac{m+h+2q}{2}} \right\} \\ &= B(n, m, q) \ . \end{aligned} \tag{3.9}$$

A similar computation yields $A(n, m, q)$ and thereby the claim of the Proposition. ■

We define

$$D_q = |\{m : \ 0 \leq m \leq n, \quad M_n - S_m = q\}| \ . \tag{3.10}$$

Then D_q is a random variable, and

$$E(D_q) = \sum_{m=0}^{n} Pr(M_n - S_m = q) \ . \tag{3.11}$$

Corollary 1. *There is a constant $c_3 > 0$ such that for every $q \geq 0$,*

$$E(D_q) \leq c_3(q+1) \ . \tag{3.12}$$

Proof. By Lemma 1, we have

$$\binom{r}{j} \leq \binom{r}{\lfloor r/2 \rfloor} \leq c_4(r+1)^{-1/2} 2^r \tag{3.13}$$

for every $r, j \geq 0$. Therefore Proposition 1 and Eq. (3.11) yield

$$E(D_q) \leq c_5(2q+1) \sum_{m=0}^{n} ((m+1)(n-m+1))^{-1/2} \leq c_3(q+1) \ . \tag{3.14}$$

Lemma 2. *If $k \equiv n(\bmod\ 2)$, $k \leq r$, then*

$$\begin{aligned} Pr(M_n = r \quad and \quad S_n = k) &= \left\{ \binom{n}{\frac{n+2r-k}{2}} - \binom{n}{\frac{n+2r+2-k}{2}} \right\} 2^{-n} \\ &= 2\ \frac{2r-k+1}{n+2r-k+2} \binom{n}{\frac{n+2r-k}{2}} 2^{-n} , \qquad (3.2) \\ Pr(M_n \leq r \quad and \quad S_n = k) &= \left\{ \binom{n}{\frac{n+k}{2}} - \binom{n}{\frac{n+2r+2-k}{2}} \right\} 2^{-n} . \qquad (3.3) \end{aligned}$$

Proof. These formulas follow from Lemma 1 in Section 7 of Chapter III of [7]. (Note that the two inequality signs in that Lemma should be reversed.) ■

Lemma 3. *For any integer $r \geq 0$,*

$$Pr(M_n = r) = \binom{n}{\lfloor \frac{n-r}{2} \rfloor} 2^{-n} . \qquad (3.4)$$

Proof. This follows from Theorem 1 of Section 7 of chapter III in [7]. It also follows from Lemma 2. ■

The main results we need are about the distribution of $M_n - S_m$ for $0 \leq m \leq n$. In particular, we need to prove that random walks do not spend much time close to their maxima. There are some results of this type in the literature (see Chapter 13 of [17]), but they apply directly only to individual excursions of a random walk. There is a beautiful result of Csáki (Theorem 13.23 of [17]) which gives the exact distribution of the number of times a random walk is at its maximum, but the method of proof does not seem to extend to give the more general result we need.

Proposition 1. *If $0 \leq m \leq n$, $q \geq 0$, then*

$$Pr(M_n - S_m = q) = A(n,m,q) + B(n,m,q) , \qquad (3.5)$$

where

$$\begin{aligned} A(n,m,q) &= 2^{-n} \binom{m}{\lfloor \frac{m+q+1}{2} \rfloor} \sum_{j=0}^{q} \binom{n-m}{\lfloor \frac{n-m+j}{2} \rfloor} , \qquad (3.6) \\ B(n,m,q) &= 2^{-n} \binom{n-m}{\lfloor \frac{n-m-q}{2} \rfloor} \sum_{j=0}^{q-1} \binom{m}{\lfloor \frac{m-q+1}{2} \rfloor + j} . \qquad (3.7) \end{aligned}$$

(We use the standard convention that $\binom{a}{b} = 0$ if $b < 0$ or $b > a$.)

3. Average running time for exact maximum

It is easy to prove a lower bound for the average running time of the form $c_1 n^{1/2}$ for some small constant $c_1 > 0$. A typical random walk has M_n of order $n^{1/2}$. If the searcher probes many fewer than $n^{1/2}$ of the S_k, there will be a stretch of many more than $n^{1/2}$ consecutive positions that she will know nothing about. Therefore she will not be able to conclude that there is no large S_k in that unexplored region of the random walk. It takes some work to make this argument rigorous, but it is not hard to do. We will not do this, since we will prove the more precise result of Theorem 2.

The proof of Theorem 2 is conceptually easy. The main idea that is exploited is that usual pictures of random walks, such as that of Fig. 1, are misleading. Since M_n is almost always of order $n^{1/2}$, graphs of S_k have scales on the horizontal and vertical axes that differ by a multiplicative factor of $n^{1/2}$. This makes the random walk look much wilder than it is. Instead, almost all random walks have only gentle rises and falls. (There are theorems, such as that of Strassen [17], which make this statement precise for infinite walks.) For example, in the random walk of Fig. 1, $\max_k |S_k - S_{k+1000}| = 134$. This means that a few probes suffice to obtain a good approximation to M_n. The average value of S_k is 0 for any k, and the average value of M_n is $\sim (2n/\pi)^{1/2}$ as $n \to \infty$. Hence the typical gap between consecutive positions that have to be probed is of order $n^{1/2}$, which yields the bound of Theorem 2. Of course, this argument is a rough heuristic only, since there are various statistical dependencies and different types of averages that are involved. However, a rigorous argument can be derived, and will be given in the rest of this section.

We first state some auxiliary results.

Lemma 1. *If $|k - n/2| \le n/4$, then*

$$\binom{n}{\lfloor n/2 \rfloor + k} = \left(\frac{2}{\pi n}\right)^{1/2} 2^n \exp(-2k^2/n + O(1/n + |k|^3/n)) \ . \tag{3.1}$$

Proof. This follows from Stirling's formula [15]. It is easiest to first derive this estimate for $k = 0$ by using the standard Stirling approximation, and then estimate the ratio $\binom{n}{\lfloor n/2 \rfloor + k} / \binom{n}{\lfloor n/2 \rfloor}$. ■

We also need some basic facts about maxima of random walks. There is a wealth of information on this topic in [17, 19]. However, all we will need and some basic formulas that can be found in Section 7 of Chapter III of [7], for example. They are easy to derive using the reflection principle.

S_{k+1} and S_{k+2}. If $S_{k+3} = S_k + 1$, it suffices to probe S_{k+2}, whereas if $S_{k+3} = S_k - 1$, we only need to probe S_{k+1}. Thus in all cases at most two of the three positions S_{k+1}, S_{k+2}, S_{k+3} need to be probed.

To prove the lower bound of Theorem 1, consider an oracle that gives answers consistent with

$$\begin{aligned} S_1 &= S_3 = S_5 = \cdots = S_{2\lceil n/2 \rceil - 1} = 1 \ , \\ S_2 &= S_4 = S_6 = \cdots = S_{2\lfloor n/2 \rfloor} = 0 \ . \end{aligned}$$

In order for the searcher to determine that $M_n \geq 1$, she has to determine the value of some S_{2k-1}. On the other hand, to make sure that $M_n < 2$, she has to know that $S_{2k} = 0$ for all k. Therefore she has to ask $\geq \lfloor n/2 \rfloor + 1$ questions.

We now prove the upper bound of Theorem 1. The first step is to prove this bound for odd n by induction. If $n = 1$, then a single probe about S_1 suffices, and so the bound is true for this case. Suppose we have shown that M_{2k-1} can be determined with k probes. To determine M_{2k+1} with $k+1$ probes, we start by asking for the value of S_{2k+1}. If $S_{2k+1} \leq -1$, then we do not need to check the value of S_{2k}, and therefore only need to find the maximum of $S_0, S_1, \ldots, S_{2k-1}$, which by the induction hypothesis can be done with $\leq k$ probes. If $S_{2k+1} \geq 1$, then we do not need to ask about S_1, as

$$M_n = \max(S_2, S_3, \ldots, S_{2k+1}) \ . \tag{2.1}$$

Now

$$\max(S_2, S_3, \ldots, S_{2k+1}) = S_{2k+1} + \max(S'_0, S'_1, \ldots, S'_{2k-1}) \ , \tag{2.2}$$

where

$$S'_0 = 0, \quad S'_j = X'_1 + \cdots + X'_k \ , \quad 1 \leq j \leq 2k-1 \ , \tag{2.3}$$

and $X'_i = -X_{2k+2-i}$. (This corresponds to reversing the walk, so it starts at S_{2k+1}, and dropping the initial two steps.) Since we know S_{2k+1} from the initial probe, we reduce to finding the maximum of a walk of $2k-1$ steps, which by induction can be done with k probes. Thus in all cases $k+1$ probes suffice if $n = 2k+1$ is odd.

If $n = 2k$ is even, we ask for S_{2k}, and then determine M_{2k-1} with $\lfloor (n-1)/2 \rfloor + 1 = k$ probes. This gives us M_{2k} in the required total of $k+1 = \lfloor n/2 \rfloor + 1$ probes.

Another application of the results of this paper answers a query posted to TheoryNet by M. Kolountzakis [14]. Kolountzakis was considering the comparative efficiency of local sequential searches versus ones that use global information. He ran extensive simulations of the following algorithm, which we will call Algorithm K. The searcher starts at $m = 0$, and knows $M_0 = S_0 = 0$. If at some time the searcher has just probed S_m, and knows M_m, she then probes S_{m+k}, where $k = M_m - S_m + 1$. (Thus she goes as far to the right as possible without danger that she will miss some S_h with $S_h > M_m$.) Kolountzakis noticed that the cost of Algorithm K was much higher than that of algorithms that were not restrained to move in this local linear way, and he asked for an estimate of this cost. In Section 6 we answer his question.

Theorem 4. *The average cost of Algorithm K is*

$$\sim (2\pi^{-1}n)^{1/2}\log n \quad \textit{as} \quad n \to \infty \ . \tag{1.5}$$

Thus Theorem 4 shows that the local linear search of Algorithm K is worse by a factor of about $\log n$ than the optimal global search. The proof of Theorem 4 follows easily from the auxiliary results developed to prove Theorem 2.

2. Worst case bounds

In this section we prove Theorem 1. We first note, though, that Theorem 1 assumes that the searcher can probe any of the S_k in any order. If we consider on-line algorithms, in which no S_j with $j < k$ can be probed once S_k has been probed, then the worst case search requires n probes. To see this, consider an adversary that responds to the searcher's probes by saying that $S_k = S_{k-1} + 1$ if $S_1, S_2, \ldots, S_{k-1}$ have all been probed, and the searcher asks about S_k. However, if $S_1, \ldots, S_{k-1}$ have been probed, but the searcher's next question is about S_{k+r} for some $r \geq 1$, the adversary will from that point on give answers consistent with $S_{k-1+2s} = S_{k-1}$, $S_{k+2s} = S_{k-2}$ for $s \geq 1$. The searcher will be unable to determine M_n without knowing S_k.

When we allow the searcher to backtrack only a bounded number of steps (i.e., to ask for S_{k-j} only for $j \leq B$ once S_k has been probed), the worst case running time decreases to $\sim c_B n$ as $n \to \infty$, where $c_B \to 1/2$ as $B \to \infty$, but $c_B > 1/2$ for all B. We do not present the full details (which involve steps similar to those used below), except for showing that if $B = 2$, at most $2n/3 + 2$ probes are needed. If we have determined $\max S_j$ for $0 \leq j \leq k$, and we have probed S_k, then as the next step we probe S_{k+3}. If $S_{k+3} = S_k \pm 3$, there is no need to probe adjacent samples than ours. It does not offer the scope that our model has for comparing the effects of allowing errors, or of searching for approximate maxima.

The random walk model we use can be thought of as a paradigm of trying to find the maximum of a largely unknown function. The limitation of the random walk increments to ± 1 can model bounds on the derivative of the function. If the function is difficult to compute, then the cost of the algorithm will be dominated by function evaluations, and so charging only for probes might be realistic. Such functions arise in a variety of contexts, such as the one listed below, or those given in [9]. In most situations the function to be maximized will have a continuous argument. That problem can be modeled by our results when M_n is to be determined approximately. (See Section 5.) What our estimates show is that often substantial savings are possible if one chooses an appropriate search strategy, first probing at a sparse and well-separated set of points, and then exploring in greater detail those regions that cannot be ruled out as being of no interest.

In some search problems it is sufficient to find the approximate maximum with high probability. However, the main results of this paper are aimed at the problem of determining the exact maximum. A basic reason for this is that the tools developed for the exact maximum search make it easy to prove other results. Another reason for this emphasis on exact results came from work on trigonometric polynomials with either 0,1 or ± 1 coefficients. Polynomials of this type that have some special properties (such as never being large, never being small, or being almost constant in absolute magnitude) have long been of interest in acoustics, analysis, spectroscopy, communications, and other fields (see [16, 18] for references). While there are some theoretical results, the best polynomials for moderate degrees are found by exhaustive searches over the full set of candidates. Since the number of polynomials to be examined grows exponentially in the degree, fast algorithms are needed. Function evaluations are slow, so it is important to minimize them. In the searches reported in [16], the algorithm was based on ideas similar to those of this paper. Polynomials examined there are primarily of the form

$$f(\theta) = \sum_{k=0}^{m} a_k \exp(ik\theta) \ , \qquad a_k = \pm 1 \ . \tag{1.4}$$

The values $|f(\theta)|$, $0 \le \theta \le 2\pi$, do not behave like the profile of a random walk $S_1, S_2, \ldots, S_n$, but they are strongly correlated for nearby values of θ, and the strategy of initially probing at a sparse set of points worked well. While the model of this paper does not apply directly to the problems explored in [16], it helps to explain and justify the method of that work.

The reason that the searcher only has to examine about $n^{1/2}$ of the S_k is that if she knows that $S_h = Z$ for some h, and after probing the k-th position she finds that $S_k < Z$, then she can conclude that

$$S_{k+r} \leq Z \quad \text{for} \quad -(Z - S_k) \leq r \leq Z - S_k \ .$$

Thus there is no need to examine S_{k+r} for $|r| \leq Z - S_k$ if we are only interested in M_n. What we are exploiting here is the strong correlation between neighboring values of S_k. (It is also this correlation that makes the problem nontrivial.) Using properties of random walks, it is shown in Section 3 that probing at a sparse set of points is almost certain to produce a value of Z that is not far from M_n. On the other hand, most of the other probes will produce values of S_k considerably smaller than Z, so the searcher can safely discard most of the range and use a simple search procedure on the small remaining set.

Theorems 1 and 2 deal with determining the exact value of M_n, and determining it correctly, with no error allowed. If we ask only for a value M'_n such that $|M'_n - M_n| \leq h$, $h = 1, 2, \ldots$, then (see Section 5) the worst case cost is close to $n/(2h)$. On the other hand, for $h = o(n^{1/2})$ as $n \to \infty$, the average cost is still $\sim c_0 n^{1/2}$.

The costs change if we allow the answers to be wrong. For example (see Section 5), if we ask for a value M'_n such that $|M'_n - M_n| \leq n^{1/2}$ holds with probability > 0.9544, then the cost is 0, as no S_k need to be probed. (The searcher gives the value $M'_n = n^{1/2}$, and this has the desired probability of being correct.) In general, the cost of obtaining a value M'_n such that $|M'_n - M_n| < \epsilon n^{1/2}$ holds with probability $\geq 1 - \delta$ is $O(1)$ for all fixed $\epsilon, \delta > 0$.

The costs are still different if we ask for the exact value of M_n, but do allow for a nonzero probability of error. In Section 4 we prove the following result.

Theorem 3. *There is an algorithm that produces a value M'_n at average cost $\leq c_2(\log n)^2$ for a certain constant $c_2 > 0$ such that $Prob(M'_n \neq M_n) \leq n^{-10}$.*

The $O((\log n)^2)$ upper bound of Theorem 3 is best possible. A proof is outlined in Section 4.

The literature on searching is immense, and we list just some of the recent references [1, 2, 4, 5, 6, 8, 9, 10, 11, 12, 13, 20, 21, 22]. None of the models used in those works is close to ours, though. The nearest is probably the model of Hajek [9], which considers a stationary Gaussian random process $(X_i : i \in Z)$ with mean 0 and $E(X_i X_j) = a^{|i-j|}$, where $0 < a < 1$. Hajek shows that to determine the maximum of $X_1, \ldots, X_n$ with positive probability takes on the order of $n(\log \log n)^{-1}$ probes. Hajek's model assumes much less correlation between is told at the beginning that $S_k = 0$ for all even k, she still has to probe all the S_k for k odd if she is to be sure that $M_n = 0$, as any odd k that had not been probed could have $S_k = 1$. Therefore for some walks it is necessary to probe $\geq \lceil n/2 \rceil$ of the S_k. Somewhat surprisingly, the $\lceil n/2 \rceil$ bound is close to best possible, so that even without being given any values of S_k beforehand, it is possible to determine M_n with $\sim n/2$ probes as $n \to \infty$. The following result is proved in Section 2.

Theorem 1. *Any algorithm that determines M_n exactly has cost $\geq \lfloor n/2 \rfloor + 1$ for some walks. There is an algorithm that determines M_n exactly at cost $\leq \lfloor n/2 \rfloor + 1$ for every walk.*

The main result of this paper is the asymptotic estimation of the average cost of determining M_n if all symmetric random walks of length n are equally likely. The average cost turns out to be of order $n^{1/2}$. Somewhat surprisingly, one can achieve an average cost that is asymptotic to the average cost of the minimal proof of M_n. By a proof we mean a set of pairs (k, S_k) which allow the searcher to deduce what M_n is. For example, if we consider the walk with $S_k = -k$ for $1 \leq k \leq n$, then revealing any single pair (k, S_k) for $k \geq n/2$ proves rigorously that $M_n = 0$. Hence for this walk, the cost of a minimal proof of M_n is 1. In general, if $S_1, \ldots, S_n$ are known, it is easy (polynomial time) to find a minimal proof.

Theorem 2. *There exists a positive constant c_0 and an algorithm that determines M_n after examining on average $\leq (c_0 + o(1))n^{1/2}$ of the S_k as $n \to \infty$. Any proof of the exact value of M_n on average costs $\geq (c_0 + o(1))n^{1/2}$ as $n \to \infty$.*

The constant c_0 is given explicitly by

$$\begin{aligned} c_0 &= 2\pi^{-1} \int_0^\infty dy \int_0^1 \frac{\exp(-y^2/(2w))dw}{w^{1/2}(1-w)^{1/2}} \int_0^1 \exp(-y^2x^2/(2-2w))dx \\ &= \pi^{1/2}2^{-1/2} \int_0^\infty y^{-1}dy \int_0^1 w^{-1/2} \exp\left(-\frac{y^2}{2w}\right) \operatorname{erf}\left(\frac{y}{(2(1-w))^{1/2}}\right) dw \ , \end{aligned} \tag{1.2}$$

where $\operatorname{erf}(x)$ is the error function [15]. Numerical integration shows that

$$c_0 = 1.1061\ldots \ . \tag{1.3}$$

A lower bound of the form $c_1 n^{1/2}$ for some small constant $c_1 > 0$ is easy to obtain, and is sketched in Section 3. The upper bound and the exact lower bound of Theorem 2 are considerably more complicated, and are presented in Section 3. The algorithm of Theorem 2 can be easily parallelized.

Search for the maximum of a random walk

A. M. Odlyzko

AT&T Bell Laboratories
Murray Hill, NJ 07974
amo@research.att.com

1. Introduction

Let S_k be the position after k steps of a symmetric random walk on the integers, starting at the origin, so that $S_k = X_1 + X_2 + \cdots + X_k$, $S_0 = 0$, where the X_j are independent and identically distributed with $\mathrm{Prob}(X_j = 1) = \mathrm{Prob}(X_j = -1) = 1/2$. We consider the problem of determining

$$M_n = \max_{0 \le k \le n} S_k \tag{1.1}$$

while minimizing the number of values of S_k that are examined. More precisely, we consider the S_k, $1 \le k \le n$, as being stored someplace, and a searcher who is given the task of determining M_n. She can ask for the exact values of any of the S_k, is charged a unit cost for each S_k that she asks for, but can do an arbitrary amount of computation with the values that have been revealed.

The problem of determining M_n does not arise directly in any application. However, it can be used to model a variety of search problems, as will be mentioned later. By considering the well-defined problem above, we can apply results about random walks to determine rigorously how the cost varies depending on the search problem that is chosen. One obtains different results depending on whether one considers worst case or average case results. Allowing even a small probability of error in the answer also has a dramatic effect on the cost of the best algorithms. For example, an estimate of M_n that is guaranteed to be correct to within some additive factor (such as $n^{1/4}$, say) is much costlier to obtain than a value for M_n that is exact almost always, but with a small probability (n^{-5}, say) is wrong.

Under some circumstances the searcher can determine M_n cheaply. For example, if she asks for S_n and is told that $S_n = \pm n$, then the search can be concluded, since the only way this can happen is if $X_1 = X_2 = \cdots = X_n = S_n/n = \pm 1$, and so $M_n = n$ if $S_n = n$ and $M_n = 0$ if $S_n = -n$. Thus for the random walks with these outcomes a single probe suffices to determine M_n. On the other hand, if the random walk moves back between 0 and -1, so that $S_k = -1$ for k odd, $S_k = 0$ for k even, the cost of proving $M_n = 0$ is high. Even if the searcher

Search for the maximum of a random walk

A. M. Odlyzko

AT&T Bell Laboratories
Murray Hill, NJ 07974
amo@research.att.com

ABSTRACT

This paper examines the efficiency of various strategies for searching in an unknown environment. The model is that of the simple random walk, which can be taken as a representation of a function with a bounded derivative that is difficult to compute. Let $X_1, X_2, \ldots$ be independent and identically distributed with $\text{Prob}(X_j = 1) = \text{Prob}\ (X_j = -1) = 1/2$, and let $S_k = X_1 + X_2 + \cdots + X_k$. Thus S_k is the position of a symmetric random walk on the line after k steps. The number of the S_k that have to be examined to determine their maximum $M_n = \max\{S_0, \ldots, S_n\}$ is $\sim n/2$ as $n \to \infty$, but that is a worst case result. Any algorithm that determines M_n with certainty must examine at least $(c_0 + o(1))n^{1/2}$ of the S_k on average for a certain constant $c_0 > 0$, if all random walks with n steps are considered equally likely. There is also an algorithm that on average examines only $(c_0 + o(1))n^{1/2}$ of the S_k to determine M_n. Different results are obtained when one allows a nonzero probability of error, or else asks only for an estimate of M_n. It is also shown that a global search (one that can ask for any value S_k at any time) for the exact maximum is faster by a factor of $\log n$ (when comparing average running times) than a linear sequential one that can skip through some values but cannot go back.

Abstract of preliminary results appeared in *Adv. Appl. Prob.* *24* (1992), 768. Extended abstract of this paper was published on pp. 336–345 of *Proc. 26-th ACM Symp. Theory Comp.*, 1994.