Chapter 12

Randomized Algorithms II – High Probability

NEW CS 473: Theory II, Fall 2015
October 6, 2015

12.1 Understanding the binomial distribution

12.1.0.1 Binomial distribution

X_n = numbers of heads when flipping a coin n times.

Claim

$\mathbf{Pr}\left[X_n = i\right] = \frac{\binom{n}{i}}{2^n}$.

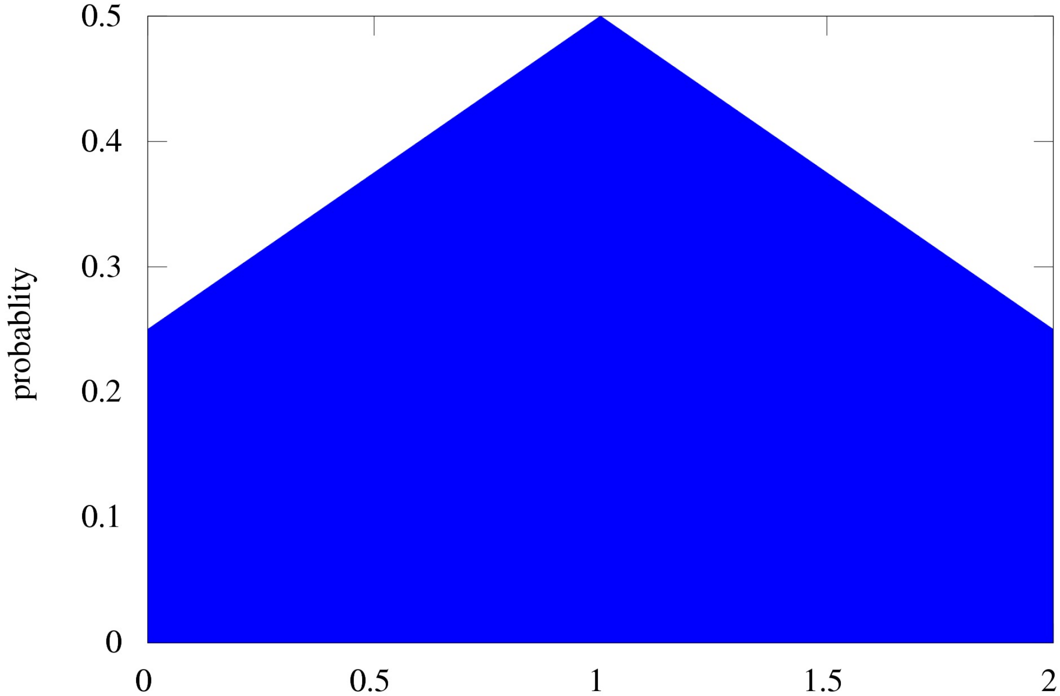
Where: $\binom{n}{k} = \frac{n!}{(n-k)!k!}$.

Indeed, $\binom{n}{i}$ is the number of ways to choose i elements out of n elements (i.e., pick which i coin flip come up heads).

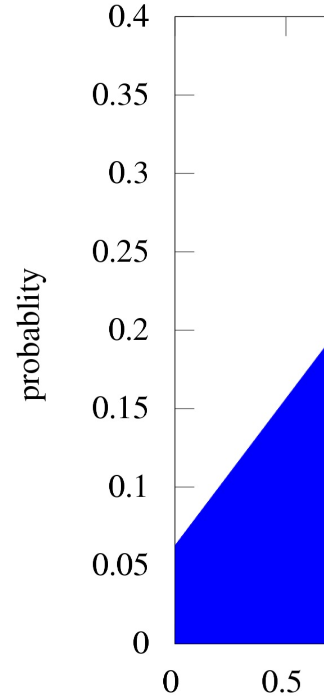
Each specific such possibility (say 0100010...) had probability $1/2^n$.

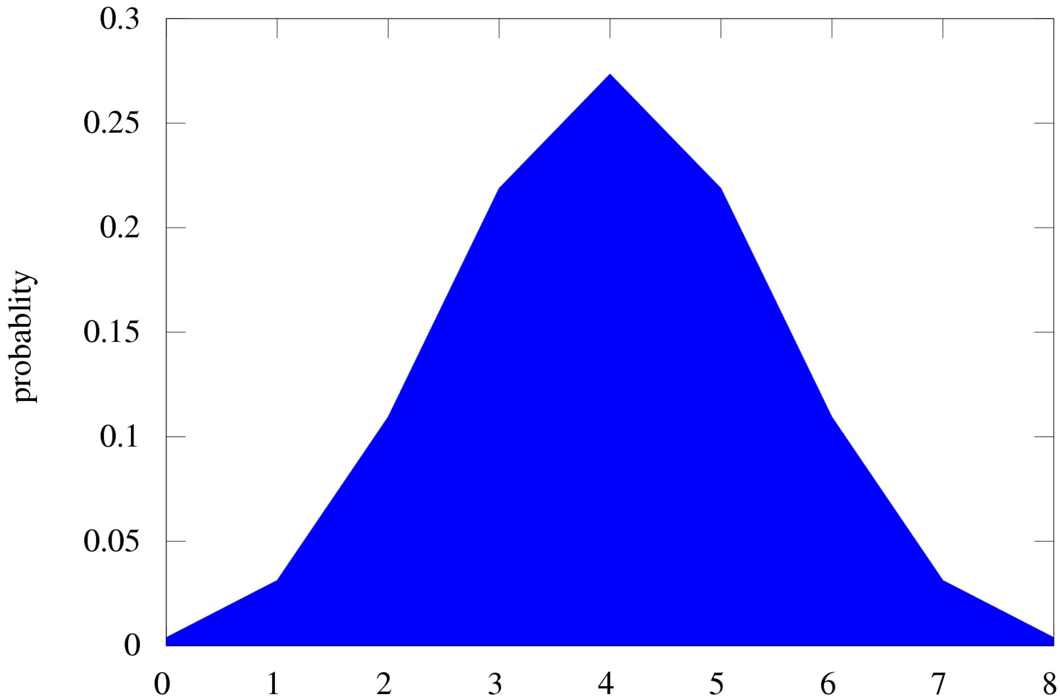
12.1.0.2 Massive randomness.. Is not that random.

Consider flipping a fair coin n times independently, head given 1, tail gives zero. How many heads? ...we get a binomial distribution.

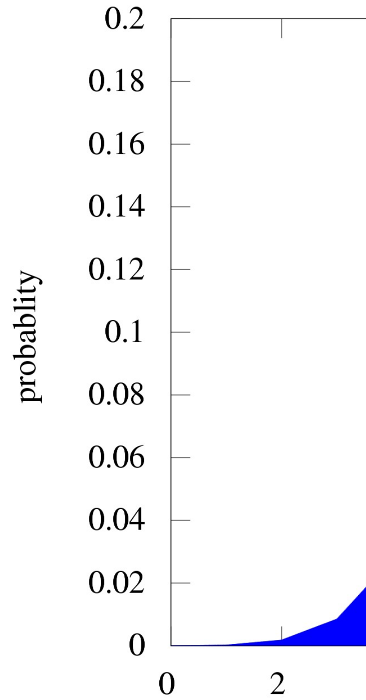

probablity
0.5
0.4
0.3
0.2
0.1
0
0.5
1
1.5
2

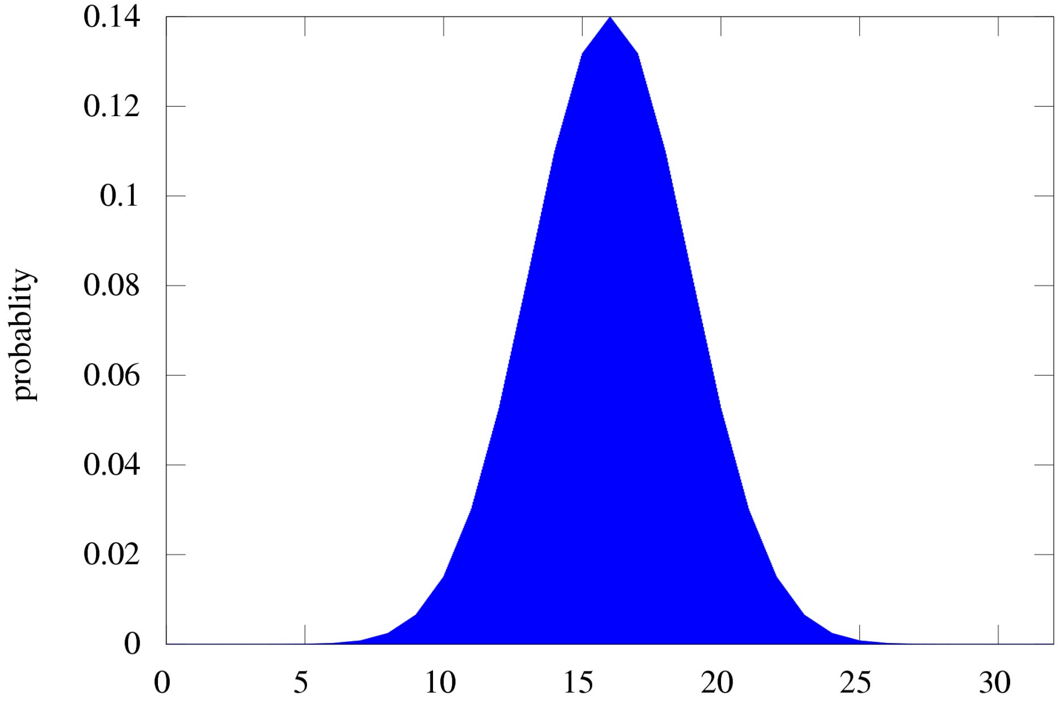
n = 2


probablity
0.4
0.35
0.3
0.25
0.2
0.15
0.1
0.05
0
0.5


probablity
0.3
0.25
0.2
0.15
0.1
0.05
0
1
2
3
4
5
6
7
8

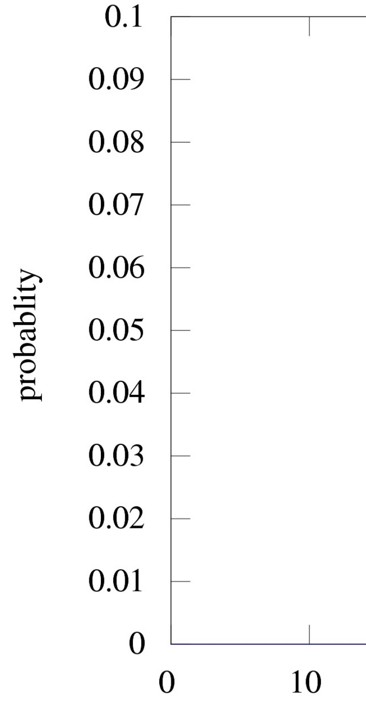
n = 8


probablity
0.2
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
2

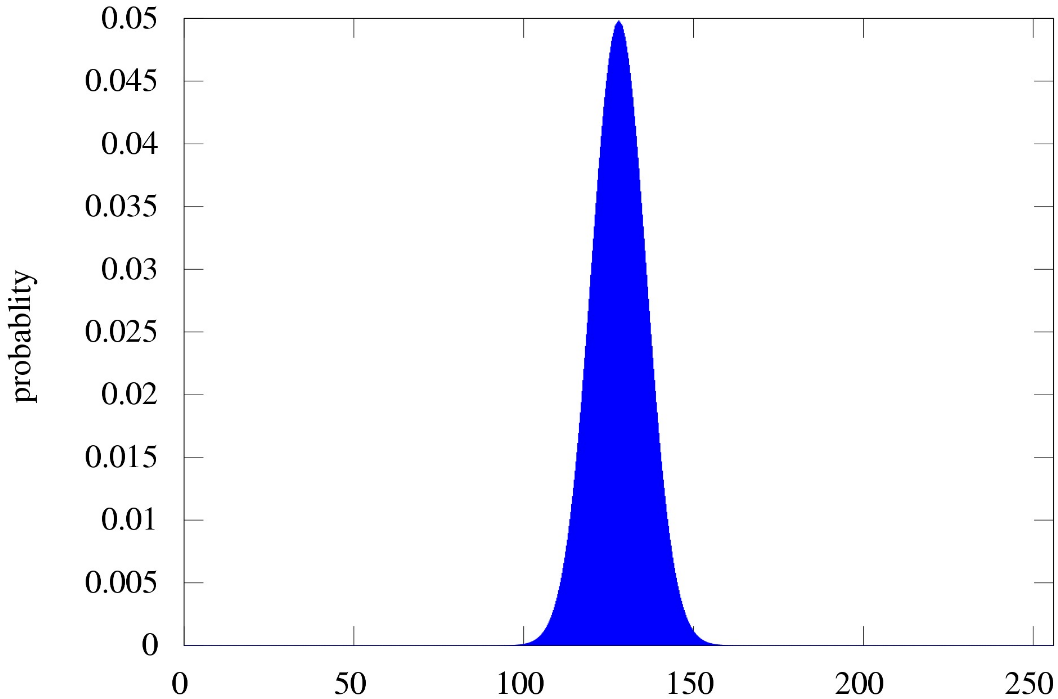


probablity
0.14
0.12
0.1
0.08
0.06
0.04
0.02
0
0
5
10
15
20
25
30

n = 32

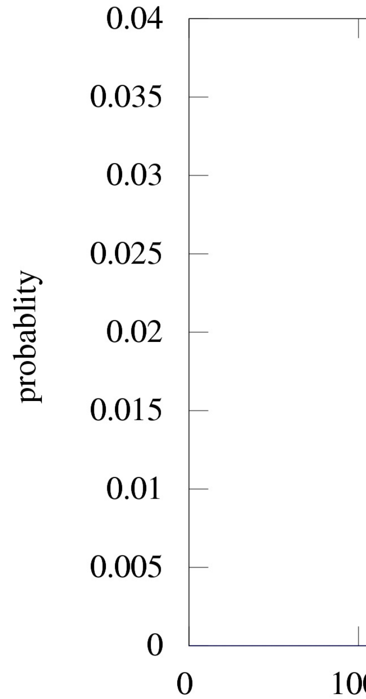


probablity
0.1
0.09
0.08
0.07
0.06
0.05
0.04
0.03
0.02
0.01
0
0
10

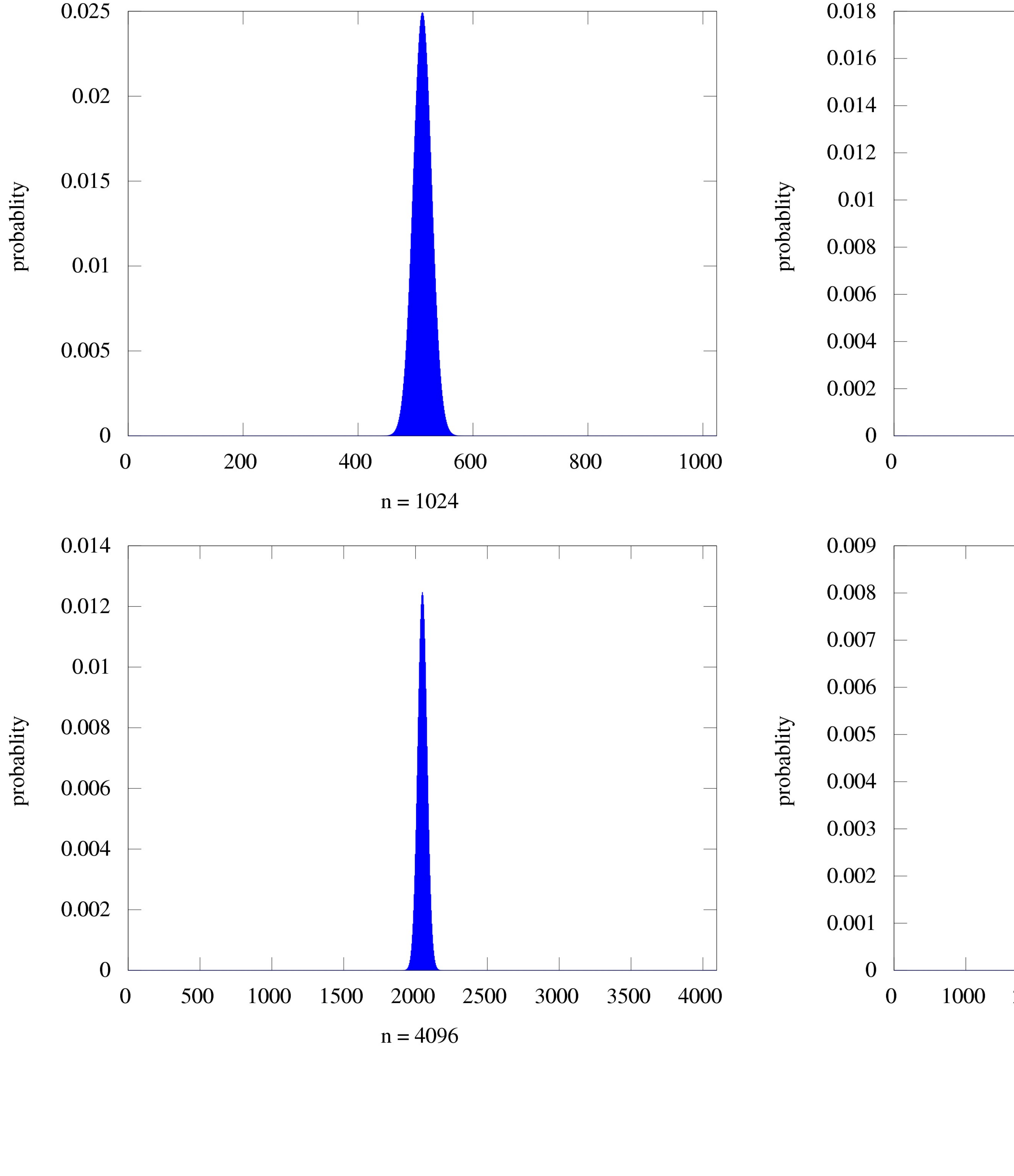


probablity
0.05
0.045
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0
50
100
150
200
250

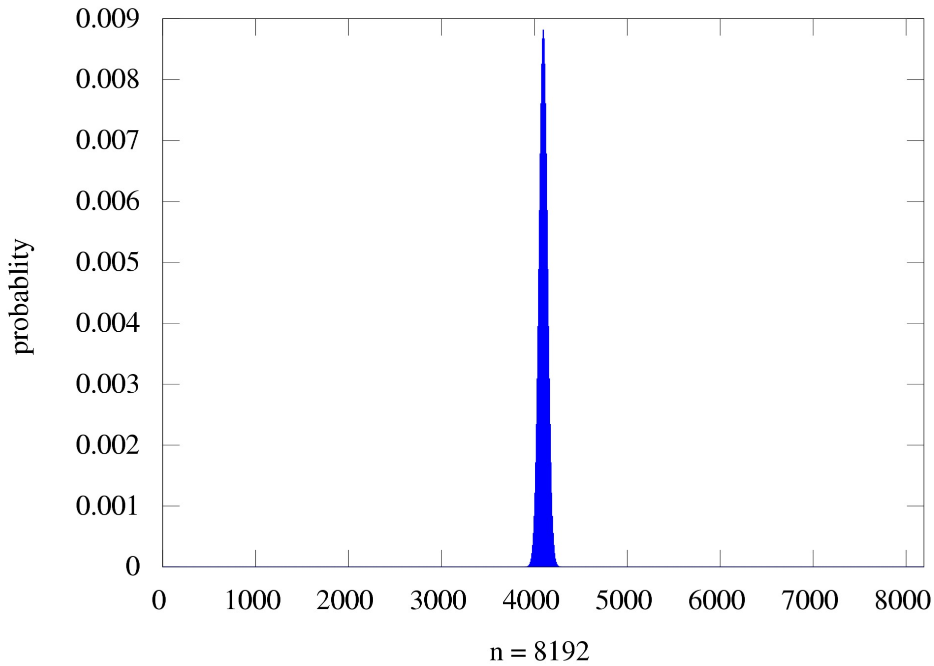
n = 256



probablity
0.04
0.035
0.03
0.025
0.02
0.015
0.01
0.005
0
0


0.025
0.02
0.015
0.01
0.005
0
probablity
0
200
400
600
800
1000
n = 1024
0.018
0.016
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
probablity
0
0.014
0.012
0.01
0.008
0.006
0.004
0.002
0
probablity
0
500
1000
1500
2000
2500
3000
3500
4000
n = 4096
0.009
0.008
0.007
0.006
0.005
0.004
0.003
0.002
0.001
0
probablity
0
1000

12.1.0.3 Massive randomness.. Is not that random.



This is known as ***concentration of mass***.

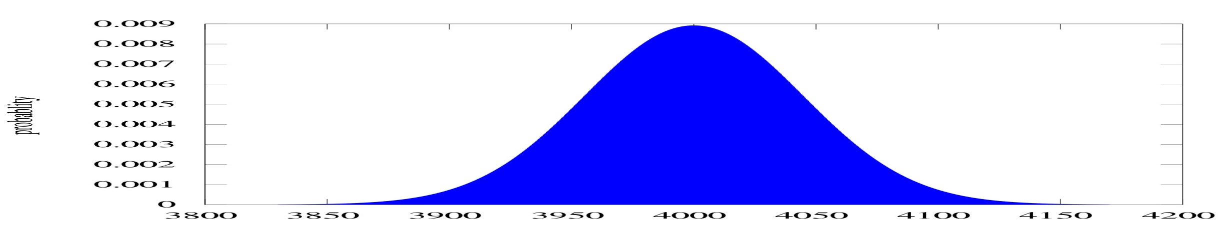
This is a very special case of the ***law of large numbers***.

12.1.1 Side note...

12.1.1.1 Law of large numbers (weakest form)...

Informal statement of law of large numbers

For n large enough, the middle portion of the binomial distribution looks like (converges to) the normal/Gaussian distribution.

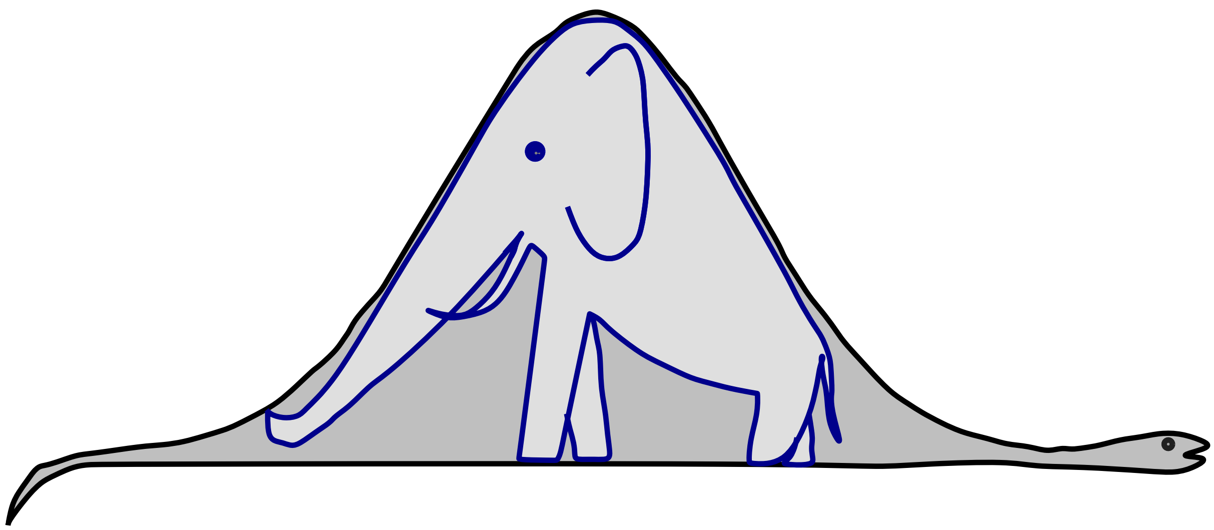


12.1.1.2 Massive randomness.. Is not that random.

Intuitive conclusion

Randomized algorithm are unpredictable in the tactical level, but very predictable in the strategic level.

12.1.1.3 What is really hiding below the Normal distribution?



Taken from Matoušek and Nešetřil [1998].

12.2 QuickSort and Treaps with High Probability

12.2.0.1 Proof of high probability of QuickSort

(A) T: n items to be sorted.
(B) $t \in T$: element.
(C) X_i: the size of subproblem S_i in ith level of recursion containing t.
(D) $X_0 = n$, and

$$\mathbf{E}\Big[X_i \;\Big|\; X_{i-1}\Big] \leq \mathbf{Pr}[\text{lucky}]\,\frac{3}{4}X_{i-1} + \mathbf{Pr}[\text{unlucky}]\,X_{i-1}$$

(E) Lucky = pivot used in S_i is in rank $\left[\frac{1}{4}\left|S_i\right|, \frac{3}{4}\left|S_i\right|\right]$
(F) $\mathbf{Pr}[\text{lucky}] = 1/2$.
(G) $\mathbf{Pr}[\text{lucky}] = 1/2$. As such...

$$\mathbf{E}\Big[X_i \;\Big|\; X_{i-1}\Big] \leq \frac{1}{2}\frac{3}{4}X_{i-1} + \frac{1}{2}X_{i-1} = \frac{7}{8}X_{i-1}.$$

12.2.0.2 Proof of high probability of QuickSort

(A) T: n items to be sorted.
(B) $t \in T$: element.
(C) X_i: the size of subproblem in ith level of recursion containing t.
(D) $X_0 = n$, and $\mathbf{E}\Big[X_i \;\Big|\; X_{i-1}\Big] \leq \frac{1}{2}\frac{3}{4}X_{i-1} + \frac{1}{2}X_{i-1} \leq \frac{7}{8}X_{i-1}$.
(E) $\forall$ random variables $\mathbf{E}\Big[X\Big] = \mathbf{E}_y\Big[\mathbf{E}\Big[X \;\Big|\; Y = y\Big]\Big]$.

(F) $\mathbf{E}\Big[X_i\Big] = \mathbf{E}_y\Big[\mathbf{E}\Big[X_i \;\Big|\; X_{i-1} = y\Big]\Big] \le \mathbf{E}_{X_{i-1}=y}\big[\frac{7}{8}y\big] = \frac{7}{8}\mathbf{E}\Big[X_{i-1}\Big] \le \big(\frac{7}{8}\big)^i \mathbf{E}[X_0] = \big(\frac{7}{8}\big)^i n.$

12.2.0.3 Proof of high probability of QuickSort

(A) $M = 8\log_{8/7} n$: $\mu = \mathbf{E}\Big[X_M\Big] \le \big(\frac{7}{8}\big)^M n \le \frac{1}{n^8} n = \frac{1}{n^7}$.

(B) **Markov's Inequality**: For a non-negative variable X, and $t > 0$, we have: $\mathbf{Pr}\Big[X \ge t\Big] \le \frac{\mathbf{E}[X]}{t}$.

(C) By Markov's inequality:

$$\mathbf{Pr}\left[\begin{array}{c} t \text{ participates} \\ > M \text{ recursive} \\ \text{calls} \end{array}\right] \le \mathbf{Pr}\Big[X_M \ge 1\Big] \le \frac{\mathbf{E}[X_M]}{1} \le \frac{1}{n^7}.$$

(D) Probability any element of input participates $> M$ recursive calls $\le n(1/n^7) \le 1/n^6$.

12.2.1 High probability via Chernoff inequality

12.2.1.1 Show that QuickSort running time is $O(n \log n)$

(A) **QuickSort** picks a pivot, splits into two subproblems, and continues recursively.
(B) Track single element in input.
(C) Game ends, when this element is alone in subproblem.
(D) Show every element in input, participates $\le 32 \ln n$ rounds (with high enough probability).
(E) $\mathcal{E}_i$: event ith element participates $> 32 \ln n$ rounds.
(F) C_{QS}: number of comparisons performed by **QuickSort**.
(G) Running time $O(C_{QS})$.
(H) Probability of failure is $\alpha = \mathbf{Pr}\Big[C_{QS} \ge 32 n \ln n\Big] \le \mathbf{Pr}[\bigcup_i \mathcal{E}_i] \le \sum_{i=1}^n \mathbf{Pr}\Big[\mathcal{E}_i\Big]$.
... by the union bound.

12.2.1.2 Show that QuickSort running time is $O(n \log n)$

(A) Probability of failure is $\alpha = \mathbf{Pr}\Big[C_{QS} \ge 32 n \ln n\Big] \le \mathbf{Pr}[\bigcup_i \mathcal{E}_i] \le \sum_{i=1}^n \mathbf{Pr}\Big[\mathcal{E}_i\Big]$.
(B) ***Union bound***: for any two events A and B: $\mathbf{Pr}[A \cup B] \le \mathbf{Pr}[A] + \mathbf{Pr}[B]$.
(C) Assume: $\mathbf{Pr}[\mathcal{E}_i] \le 1/n^3$.
(D) Bad probability... $\alpha \le \sum_{i=1}^n \mathbf{Pr}\Big[\mathcal{E}_i\Big] \le \sum_{i=1}^n \frac{1}{n^3} = \frac{1}{n^2}$.
(E) $\implies$ **QuickSort** performs $\le 32 n \ln n$ comparisons, w.h.p.
(F) $\implies$ **QuickSort** runs in $O(n \log n)$ time, with high probability.

12.2.2 Proving that an element participates in small number of rounds

12.2.3 Proving that an element...

12.2.3.1 ... participates in small number of rounds.

(A) n: number of elements in input for **QuickSort**.
(B) x: Arbitrary element x in input.
(C) S_1: Input.

(D) S_i: input to ith level recursive call that include x.
(E) x ***lucky*** in jth iteration, if balanced split...
$|S_{j+1}| \leq (3/4)\,|S_j|$ and $|S_j \setminus S_{j+1}| \leq (3/4)\,|S_j|$
(F) $Y_j = 1 \iff x$ lucky in jth iteration.
(G) $\mathbf{Pr}\Big[Y_j\Big] = \frac{1}{2}$.
(H) **Observation**: $Y_1, Y_2, \ldots, Y_m$ are independent variables.
(I) x can participate $\leq \rho = \log_{4/3} n \leq 3.5 \ln n$ rounds.
(J) ...since $|S_j| \leq n(3/4)^{\#\text{ of lucky iteration in}1\ldots j}$.
(K) If ρ lucky rounds in first k rounds $\implies |S_k| \leq (3/4)^\rho n \leq 1$.

12.2.4 Proving that an element...

12.2.4.1 ... participates in small number of rounds.

(A) Brain reset!
(B) Q: How many rounds x participates in = how many coin flips till one gets ρ heads?
(C) A: In expectation, 2ρ times.

12.2.5 Proving that an element...

12.2.5.1 ... participates in small number of rounds.

(A) Assume the following:

Lemma 12.2.1. *In M coin flips:* $\mathbf{Pr}[\#\ \textit{heads} \leq M/4] \leq \exp(-M/8)$.

(B) Set $M = 32 \ln n \geq 8\rho$.
(C) $\mathbf{Pr}[Y_j = 0] = \mathbf{Pr}[Y_j = 1] = 1/2$.
(D) $Y_1, Y_2, \ldots, Y_M$ are independent.
(E) $\implies$ probability $\leq \rho \leq M/4$ ones in $Y_1, \ldots, Y_M$ is

$$\leq \exp\left(-\frac{M}{8}\right) \leq \exp(-\rho) \leq \frac{1}{n^3}.$$

(F) $\implies$ probability x participates in M recursive calls of **QuickSort** $\leq 1/n^3$.

12.2.6 Proving that an element...

12.2.6.1 ... participates in small number of rounds.

(A) n input elements. Probability depth of recursion in **QuickSort** $> 32 \ln n$ is $\leq (1/n^3) * n = 1/n^2$.
(B) Result:

Theorem 12.2.2. *With high probability (i.e., $1 - 1/n^2$) the depth of the recursion of* **QuickSort** *is $\leq 32 \ln n$. Thus, with high probability, the running time of* **QuickSort** *is $O(n \log n)$.*

(C) Same result holds for **MatchNutsAndBolts**.

12.3 Chernoff inequality

12.3.0.1 Preliminaries

(A) X, Y: Random variables are *independent* if $\forall x, y$:

$$\mathbf{Pr}\Big[(X = x) \cap (Y = y)\Big] = \mathbf{Pr}\Big[X = x\Big] \cdot \mathbf{Pr}\Big[Y = y\Big].$$

(B) The following is easy to prove:

Claim 12.3.1. *If X and Y are independent*
$\implies$ $\mathbf{E}[XY] = \mathbf{E}[X]\,\mathbf{E}[Y]$.
$\implies$ *$Z = e^X$ and $W = e^Y$ are independent.*

12.3.0.2 Chernoff inequality

Theorem 12.3.2 (Chernoff inequality). *$X_1, \ldots, X_n$: n independent random variables, such that $\mathbf{Pr}[X_i = 1] = \mathbf{Pr}[X_i = -1] = \frac{1}{2}$, for $i = 1, \ldots, n$. Let $Y = \sum_{i=1}^{n} X_i$. Then, for any $\Delta > 0$, we have*

$$\mathbf{Pr}\Big[Y \geq \Delta\Big] \leq \exp\big(-\Delta^2/2n\big).$$

12.3.0.3 Proof of Chernoff inequality

Fix arbitrary $t > 0$:

$$\begin{aligned}\mathbf{Pr}\Big[Y \geq \Delta\Big] &= \mathbf{Pr}\Big[tY \geq t\Delta\Big] = \mathbf{Pr}\Big[\exp(tY) \geq \exp(t\Delta)\Big] \\ &\leq \frac{\mathbf{E}\Big[\exp(tY)\Big]}{\exp(t\Delta)},\end{aligned}$$

12.3.1 Proof of Chernoff inequality

12.3.1.1 Continued...

$$\begin{aligned}\mathbf{E}\Big[\exp(tX_i)\Big] &= \frac{1}{2}e^t + \frac{1}{2}e^{-t} = \frac{e^t + e^{-t}}{2} \\ &= \frac{1}{2}\left(1 + \frac{t}{1!} + \frac{t^2}{2!} + \frac{t^3}{3!} + \cdots\right) \\ &\quad + \frac{1}{2}\left(1 - \frac{t}{1!} + \frac{t^2}{2!} - \frac{t^3}{3!} + \cdots\right) \\ &= 1 + \frac{t^2}{2!} + + \cdots + \frac{t^{2k}}{(2k)!} + \cdots .\end{aligned}$$

However: $(2k)! = k!(k+1)(k+2)\cdots 2k \geq k!2^k$.

$$\mathbf{E}\Big[\exp(tX_i)\Big] = \sum_{i=0}^{\infty} \frac{t^{2i}}{(2i)!} \leq \sum_{i=0}^{\infty} \frac{t^{2i}}{2^i(i!)} =\leq \sum_{i=0}^{\infty} \frac{1}{i!}\left(\frac{t^2}{2}\right)^i =\leq \exp\left(\frac{t^2}{2}\right).$$

$$\mathbf{E}\Big[\exp(tY)\Big] = \mathbf{E}\left[\exp\left(\sum_i tX_i\right)\right] = \mathbf{E}\left[\prod_i \exp(tX_i)\right] = \prod_{i=1}^{n} \mathbf{E}\Big[\exp(tX_i)\Big] \le \prod_{i=1}^{n} \exp\left(\frac{t^2}{2}\right) = \le \exp\left(\frac{nt^2}{2}\right).$$

$$\mathbf{Pr}\Big[Y \ge \Delta\Big] \le \frac{\mathbf{E}\Big[\exp(tY)\Big]}{\exp(t\Delta)} \le \frac{\exp\left(\frac{nt^2}{2}\right)}{\exp(t\Delta)} = \exp\left(\frac{nt^2}{2} - t\Delta\right).$$

Set $t = \Delta/n$:

$$\mathbf{Pr}\Big[Y \ge \Delta\Big] \le \exp\left(\frac{n}{2}\left(\frac{\Delta}{n}\right)^2 - \frac{\Delta}{n}\Delta\right) = \exp\left(-\frac{\Delta^2}{2n}\right).$$

■

12.3.2 Chernoff inequality...

12.3.2.1 ...what it really says

By theorem:

$$\mathbf{Pr}\Big[Y \ge \Delta\Big] = \sum_{i=\Delta}^{n} \mathbf{Pr}\Big[Y = i\Big] = \sum_{i=n/2+\Delta/2}^{n} \frac{\binom{n}{i}}{2^n} \le \exp\left(-\frac{\Delta^2}{2n}\right),$$

12.3.3 Chernoff inequality...

12.3.3.1 symmetry

Corollary 12.3.3. *Let $X_1, \ldots, X_n$ be n independent random variables, such that $\mathbf{Pr}[X_i = 1] = \mathbf{Pr}[X_i = -1] = \frac{1}{2}$, for $i = 1, \ldots, n$. Let $Y = \sum_{i=1}^{n} X_i$. Then, for any $\Delta > 0$, we have*

$$\mathbf{Pr}\Big[|Y| \ge \Delta\Big] \le 2\exp\left(-\frac{\Delta^2}{2n}\right).$$

12.3.3.2 Chernoff inequality for coin flips

$X_1, \ldots, X_n$ be n independent coin flips, such that $\mathbf{Pr}[X_i = 1] = \mathbf{Pr}[X_i = 0] = \frac{1}{2}$, for $i = 1, \ldots, n$. Let $Y = \sum_{i=1}^{n} X_i$. Then, for any $\Delta > 0$, we have

$$\mathbf{Pr}\left[\frac{n}{2} - Y \ge \Delta\right] \le \exp\left(-\frac{2\Delta^2}{n}\right)$$

$$\text{and} \quad \mathbf{Pr}\left[Y - \frac{n}{2} \ge \Delta\right] \le \exp\left(-\frac{2\Delta^2}{n}\right).$$

In particular, we have $\mathbf{Pr}\left[\left|Y - \frac{n}{2}\right| \ge \Delta\right] \le 2\exp\left(-\frac{2\Delta^2}{n}\right)$.

Note: Variables $X_i \in \{0, 1\}$. Previous slide $X_i \in \{-1, 1\}$ (different result!).

12.3.3.3 The special case we needed

Lemma 12.3.4. *In a sequence of M coin flips, the probability that the number of ones is smaller than $L \le M/4$ is at most* $\exp(-M/8)$.

Proof: Let $Y = \sum_{i=1}^{m} X_i$ the sum of the M coin flips. By the above corollary, we have:

$$\mathbf{Pr}\Big[Y \le L\Big] = \mathbf{Pr}\left[\frac{M}{2} - Y \ge \frac{M}{2} - L\right] = \mathbf{Pr}\left[\frac{M}{2} - Y \ge \Delta\right],$$

where $\Delta = M/2 - L \ge M/4$. Using the above Chernoff inequality, we get $\mathbf{Pr}\Big[Y \le L\Big] \le \exp\Big(-\frac{2\Delta^2}{M}\Big) \le \exp(-M/8)$. ■

12.4 The Chernoff Bound — General Case

12.4.1 The Chernoff Bound

12.4.1.1 The general problem

Problem 12.4.1. Let $X_1, \dots X_n$ be n independent Bernoulli trials, where

$$\mathbf{Pr}\Big[X_i = 1\Big] = p_i \qquad \text{and} \qquad \mathbf{Pr}\Big[X_i = 0\Big] = 1 - p_i,$$

and let denote

$$Y = \sum_i X_i \qquad \mu = \mathbf{E}[Y].$$

Question: what is the probability that $Y \ge (1 + \delta)\mu$.

12.4.2 The Chernoff Bound

12.4.2.1 The general case

Theorem 12.4.2 (Chernoff inequality). *For any $\delta > 0$,*

$$\mathbf{Pr}\Big[Y > (1+\delta)\mu\Big] < \left(\frac{e^\delta}{(1+\delta)^{1+\delta}}\right)^\mu.$$

Or in a more simplified form, for any $\delta \le 2e - 1$,

$$\mathbf{Pr}\Big[Y > (1+\delta)\mu\Big] < \exp\big(-\mu\delta^2/4\big),$$

and

$$\mathbf{Pr}\Big[Y > (1+\delta)\mu\Big] < 2^{-\mu(1+\delta)},$$

for $\delta \ge 2e - 1$.

12.4.2.2 Theorem

Theorem 12.4.3. *Under the same assumptions as the theorem above, we have*

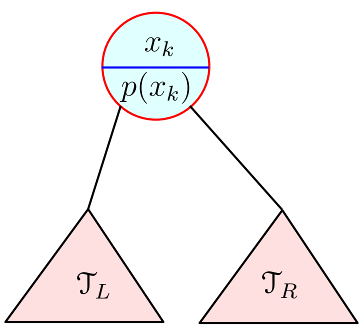
$$\mathbf{Pr}\Big[Y < (1-\delta)\mu\Big] \leq \exp\left(-\mu\frac{\delta^2}{2}\right).$$

12.5 Treaps

12.5.0.1 Balanced binary search trees...

(A) Work usually by storing additional information.
(B) Idea: For every element x inserted
randomly choose ***priority*** $p(x) \in [0,1]$.
(C) $X = \{x_1, \ldots, x_n\}$
priorities: $p(x_1), \ldots, p(x_n)$.
(D) x_k: lowest priority in X.
(E) Make x_k the root.
(F) partition X in the natural way:
  (A) L: set of all the numbers smaller than x_k in X, and
  (B) R: set of all the numbers larger than x_k in X.

12.5.0.2 Treaps



Continuing recursively, we have:
  (A) L: set of all the numbers smaller than x_k in X, and
  (B) R: set of all the numbers larger than x_k in X.

Definition 12.5.1. Resulting tree a ***treap***.
Tree over the elements, and a heap over the priorities; that is,
TREAP = TREE + HEAP.

12.5.0.3 Treaps continued

Lemma 12.5.2. *S: n elements.*
Expected depth of treap $\mathcal{T}$ for S is $O(\log(n))$.
Depth of treap $\mathcal{T}$ for S is $O(\log(n))$ w.h.p.

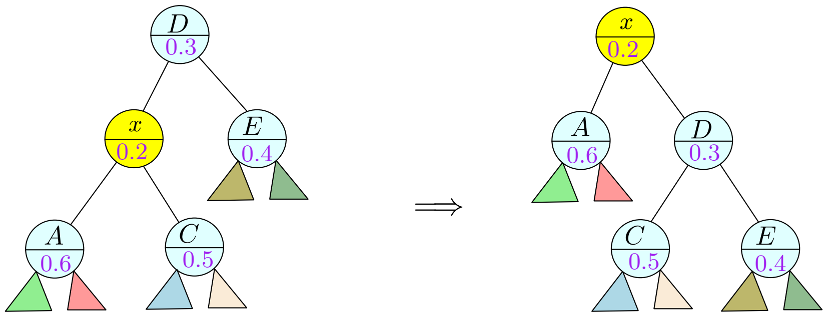
Proof: **QuickSort**... ■

12.5.1 Operations

12.5.1.1 Treaps - implementation

Observation 12.5.3. *Given n distinct elements, and their (distinct) priorities, the treap storing them is uniquely defined.*

12.5.1.2 Rotate right...



12.5.1.3 Insertion
12.5.1.4 Treaps – insertion

(A) x: an element x to insert.
(B) Insert it into $\mathcal{T}$ as a regular binary tree.
(C) Takes $O(\text{height}(\mathcal{T}))$.
(D) x is a leaf in the treap.
(E) Pick priority $p(x) \in [0, 1]$.
(F) Valid search tree,.. but priority heap is broken at x.
(G) Fix priority heap around x.

12.5.1.5 Fix treap for a leaf x...

```
RotateUp(x)
   y ← parent(x)
   while p(y) > p(x) do
         if y.left_child = x then
               RotateRight(y)
         else
               RotateLeft(y)
         y ← parent(x)
```

Insertion takes $O(\text{height}(\mathcal{T}))$.

12.5.1.6 Treaps – deletion

(A) Deletion is just an insertion done in reverse.
(B) x: element to delete.
(C) Set $p(x) \leftarrow +\infty$,
(D) rotate x down till its a leaf.
(E) Rotate so that child with lower priority becomes new parent.
(F) x is now leaf – deleting is easy...

12.5.1.7 Split

(A) x: element stored in treap $\mathcal{T}$.
(B) split $\mathcal{T}$ into two treaps – one treap $\mathcal{T}_{\leq x}$ and treap $\mathcal{T}_{>}$ for all the elements larger than x.
(C) Set $p(x) \leftarrow -\infty$,

(D) fix priorities by rotation.
(E) x item is now the root.
(F) Splitting is now easy....
(G) Restore x to its original priority. Fix by rotations.

12.5.1.8 Meld

(A) $\mathcal{T}_L$ and $\mathcal{T}_R$: treaps.
(B) all elements in $\mathcal{T}_L$ ¡ all elements in $\mathcal{T}_R$.
(C) Want to merge them into a single treap...

12.5.1.9 Treap – summary

Theorem 12.5.4. *Let $\mathcal{T}$ be an empty treap, after a sequence of $m = n^c$ insertions, where c is some constant.*

d: arbitrary constant.

The probability depth $\mathcal{T}$ ever exceed $d \log n$ is $\leq 1/n^{O(1)}$.

A treap can handle insertion/deletion in $O(\log n)$ time with high probability.

12.5.1.10 Proof

Proof: (A) $\mathcal{T}_1, \ldots, \mathcal{T}_m$: sequence of treaps.
(B) $\mathcal{T}_i$ is treap after ith operation.
(C) $\alpha_i = \mathbf{Pr}\Big[\text{depth}(\mathcal{T}_i) > tc' \log n\Big] = \mathbf{Pr}\Big[\text{depth}(\mathcal{T}_i) > c't\Big(\frac{\log n}{\log |\mathcal{T}_i|}\Big) \cdot \log |\mathcal{T}_i|\Big] \leq \frac{1}{n^{O(1)}}$,
(D) Use union bound...

12.5.1.11 Bibliographical Notes

(A) Chernoff inequality was a rediscovery of Bernstein inequality.
(B) ...published in 1924 by Sergei Bernstein.
(C) Treaps were invented by Siedel and Aragon Seidel and Aragon [1996].
(D) Experimental evidence suggests that Treaps performs reasonably well in practice see Cho and Sahni [2000].
(E) Old implementation of treaps I wrote in `C` is available here: `http://valis.cs.uiuc.edu/blog/?p=6060`.

Bibliography

S. Cho and S. Sahni. A new weight balanced binary search tree. *Int. J. Found. Comput. Sci.*, 11(3): 485–513, 2000.

J. Matoušek and J. Nešetřil. *Invitation to Discrete Mathematics.* Oxford Univ Press, 1998. ISBN 0198502079. URL `http://www4.oup.co.uk/isbn/0-19-850208-7`.

R. Seidel and C. R. Aragon. Randomized search trees. *Algorithmica*, 16:464–497, 1996.